Classifying the Season Using Street Fashion Data

Springboard Data Science Career Track Program

Maia Paddock

Capstone Project 2 - Final Report

Introduction

The fashion industry struggles from the unique problem that all of the merchandise is inherently visual. Unlike consumers buying electronics or home goods, the details on the performance and use of the items is frequently not as important as the aesthetic appearance of the item. Additionally, these aesthetics can be difficult to classify, as trends and consumer tastes change with time, location, and with different consumers. Therefore, the fashion industry has embraced machine learning as a way to tackle these complicated trends.

Manual classification of images is a time intensive task. Therefore, collecting and tagging datasets of fashion images with associated metadata of any type can be time-consuming. Many companies rely on user input to give them the metadata needed to train models for recommendation engines or to classify clothing.The Street Fashion Data Set collected for the paper Understanding Fashion Trends from Street Photos via Neighbor-Constrained Embedding Learning[1] circumvented this challenge by taking data from the website Chictopia[2]. The data from this website is well suited for machine learning applications. Each image on this website visually similar, consisting of a full body image of a single person standing in front of a (generally urban) background. Additionally, each image comes with associated metadata on the outfit. Some of this data is created by the users themselves, such as tags describing the fashion pictured in the image, and some of the data is generated by the website, such as the time the image was uploaded. This collection of almost 300,000 images and their associated metadata is large enough to conduct extensive analysis.

The metadata of the street photos includes the user name, the name of the associated photo, the location, the date it was uploaded, and two categories of user-added data: tags and styles. Styles is the more abstract of these two categories, including entries such as “classic”, “everyday”, “fall”, “chic”, “comfortable”, “beach”, and “trendy.” These style descriptors give an insight into the mind of the users and how they would classify their own outfits. The tags on the outfits are more concrete descriptors, including brand names, colors, and types of clothing, such as “dress” or “heels.”

There are many different analyses that can be performed with this dataset. The location data suffers from not being dense or evenly distributed, meaning that it would be difficult to create a classification model to predict the location of an image based on the properties of the image and other metadata. However, exploratory data analysis could still show interesting trends related to

[1] Understanding Fashion Trends from Street Photos via Neighbor-Constrained Embedding Learning by X. Gu, Y. Wong, P. Peng, L. Shou, G. Chen, M. Kankanhalli http://doi.org/10.1145/3123266.3123441
[2] http://www.chictopia.com/

location. The dataset covers a period of 8 years, from 2008 through 2015. While the data is most likely not dense enough across time to be used for any time projection, looking for any cyclic behaviors in the tags and styles could provide helpful information. The most interesting and useful categories of metadata on this dataset are the styles and tags. These two types of descriptors for the images provide a wealth of possibilities for analysis of these images and the dataset as a whole.

This dataset lends itself to two classification problems: the classification of the images by tags and by styles. Image recognition methods were tested to create a model which returns tags for each image based on the image itself. Tags are mainly colors and items of clothing, and therefore with the correct model theoretically could be found using only the images. However, this analysis ran into troubles given the size of the images and the amount of data needed to create a full deep learning model. Less than 1/25th of the available images were able to be used, leading to a weak multilabel classification with high recall but no precision.

The initial idea was to build from this analysis, using a combination of tags and the images themselves to create another classification model to return the overall style of the image. This model would both the visual data and the pre-existing tags on the image, hoping to use the correlations between the frequency of different tags and styles. For example, the tag “sandal” would likely be more correlated with the styles “summer” and “casual” than “winter” or “formal.” However, given the same issue with the images, this analysis had to be adjusted so that only the tags were used to predict the styles. The final model was a multiclass classifier finding the difference between clothes identified as belonging to the four different seasons.

Despite the setbacks, this analysis still provided a trained predictive model for classifying fashion. This model could form the basis of recommendation systems for fashion retail by assisting in the feature classification of clothes. If both classification models had performed successfully, they could have been combined into a pipeline that automatically returns tags and styles for any given image. The end goal of these models was to provide both concrete and abstract features of clothes, potentially creating a broad amount of easy to manipulate, human-readable data from only images. While the final model had to be altered from this original plan, it still could prove useful for recommendation engines. Seasons play a huge roll in selling fashion, as clothes needs to be displayed and advertised to customers before the season in which it is relevant. Therefore, a this season classification model could be inserted into a clothing recommendation pipeline to make sure the proper season is being advertised to the customers and improve the response to the suggestions.

Approach

The Data

Data was obtained from the street fashion dataset found at https://zenodo.org/record/833051. The complete files contained 293,105 data points, originally crawled from the website Chictopia. There are approximately 27 Gb of image data, split into 27 files of 1 Gb each. The metadata for all of the images is contained in a separate csv file, with the image names as one of the columns to facilitate matching the correct row to the corresponding image. Additionally, a set of pre-cleaned data that was used in the paper referenced above is available for download. However, I did not choose to use this pre-cleaned data set as it would limit my investigation.

The metadata was cleaned first, before the images were addressed. The first step in cleaning the metadata was the remove all null entries from the users, picture name, location, and time categories. This removed approximately 9000 entries that had null entries in one of those four categories. Next, the time column was converted to a pandas datetime object in order to make working with times less complicated. The times in this column included only day, month, and year, therefore time zones were not considered or added to the data based on location.

The next challenge was the clean and separate the tags and styles categories. The original metadata included all of the tags in a single column and all of the styles in another single column. These had to be split into different columns for each of the separate styles and tags. For example, an original entry in the styles column was “chic, beach, summer,” which had to be split into three columns, style_0, style_1, and style_2, each of which held one of the words. Almost all of the entries had at least two styles listed, and approximately two-thirds of the data had three styles listed. None of the entries had more than three styles. To standardize the styles, all styles with more than one word were removed from the data.

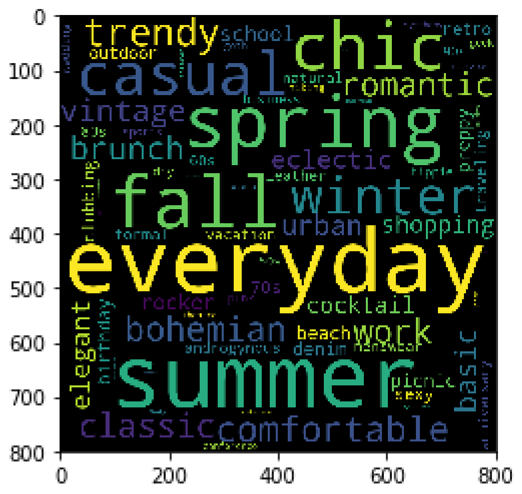
Tags were far more difficult to clean for multiple reasons. While most entries included three styles, the number of tags on an entry ranged from one to seventeen. Initially, these were split from a single tag column into columns tags_0 through tags_16. Less than 1.5% of the total non-null tag data was included in the last seven columns, therefore these were dropped to help with processing time when looping through all of the data. Adding another challenge to cleaning the tag data, the tags were less standard and more variable than the styles. Brand names were included in the tags, leading to many multiple-word entries and more variable ways of expressing the same idea. The brand names had to be removed from the data because they would make the task of training a classifier based on the images much more difficult. The goal is a classifier that can tag colors and types of clothing. Therefore, for example, having different categories for “dress”, “ASOS dress”, and “Forever 21 dress” would add more confusion to the model and take away information that could be used to better classify dresses. Initially, I attempted to use the package Spacy to perform Named Entity Recognition (NER) and strip the brand names from each of the tags. However, I was only able to get an NER that could recognize and remove less than half of the brand names. By looking at the first few hundred entries in the dataframe, I was able to determine that a far more effective way to remove the brand names would be to simply keep only the last word of any phrase. Therefore, entries such as “Forever 21 dress” would be simplified to “dress”, keeping only the data that I wanted. There are a few drawbacks to this method, but overall it was much more successful than using NER. One drawback is that it did not catch all brand names, because some brand names were tagged as a single word. Additionally, this method got rid of some useful data, for example “purple dress” was simplified to “dress”, removing the color information that could be useful as a tag. The way to fix this would have been to split each entry down to a single word, creating more tag columns, and then applying an effective NER to remove the brand names. However, as I could not find an NER that could remove enough brand names to make this a viable option, I settled with keeping only the final word of each tag.

Exploratory Data Analysis

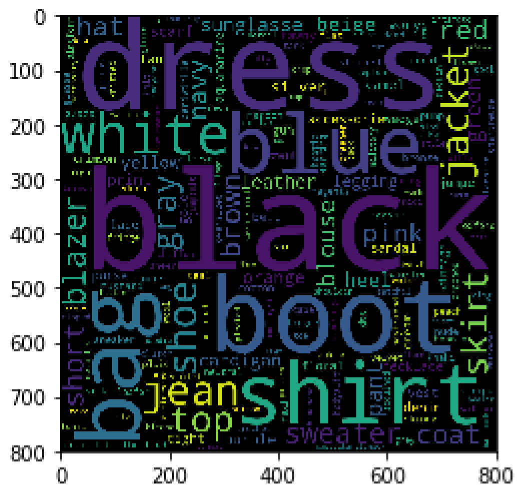
After these cleaning steps were completed, I saved the data to a new separate csv file that could be loaded to perform exploratory data analysis and statistical analysis. Just as with the data cleaning, I knew that performing EDA on the tags and styles would be the most difficult part of the process. Therefore, I started with looking at the user names and locations of the pictures. There were 3150 unique locations listed in the data, almost 900 of which are only mentioned once in the data. The majority of the locations are listed less than 12 times. Therefore, it is unlikely that locations will be useful in the model as the data is too sparse. The most listed location was unsurprisingly Los Angeles, CA, with almost 16000 entries.

There are 4976 unique users included in the data, each uploading a median of 20 images. A small number of users were responsible for a large amount of the data, with approximately 100 users uploading over 1000 images each. The most active user uploaded 8672 images. While it is possible that these users are individuals, it is likely that some of these users were in fact fashion photographers who uploaded photos of people other than themselves. The inclusion of fashion photographers in the data might influence the correlation between the user and the styles of their clothing.

Analysing the tags and the styles was a multistep processes. The first part of the process was to understand the compositing of the tags and styles using a word cloud. These word clouds allowed me to see that the data was in fact not fully cleaned. There were in fact some entries that contained the string "nan" instead of just being set to none, and these had to be removed. Next, I had to figure out how many of each of these tags and styles could be used for the analysis. The median value for the number of times a tag appeared was one. This clearly is not useful for a classification problem, where multiple instances of the tag would be necessary in order to train and test the classifier. Therefore, I made the somewhat arbitrary decision to cut on tags that appeared more than 100 times. This cutoff number can be adjusted during training to include more or less tags based on the performance of the classifier. Given this cutoff, there are 258 tags that appear more than 100 times. Any tags that did not correspond to these 258 words were removed from the data, and the data was reshaped so that there were no null tag values and it could be used in a classifier. The same process was repeated for the styles, leaving 68 styles with more than 100 occurrences. The word clouds for the styles and tags are shown below.



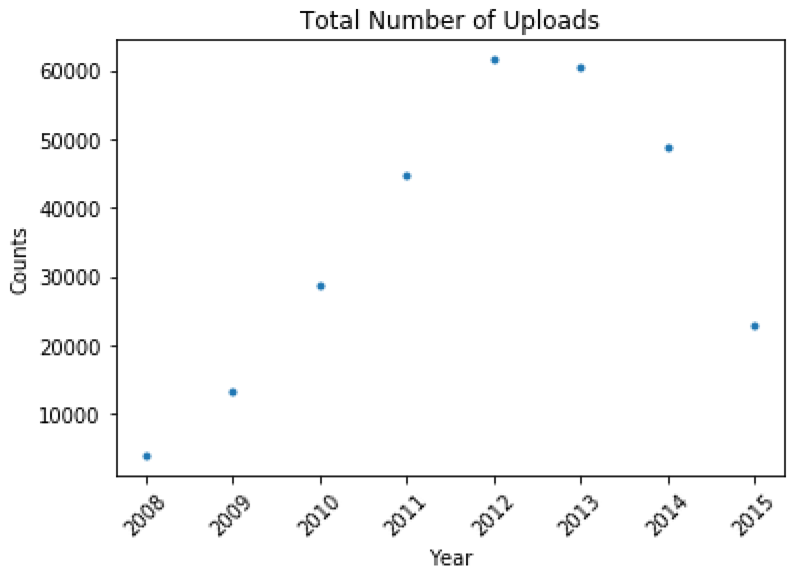
Word Cloud of Styles

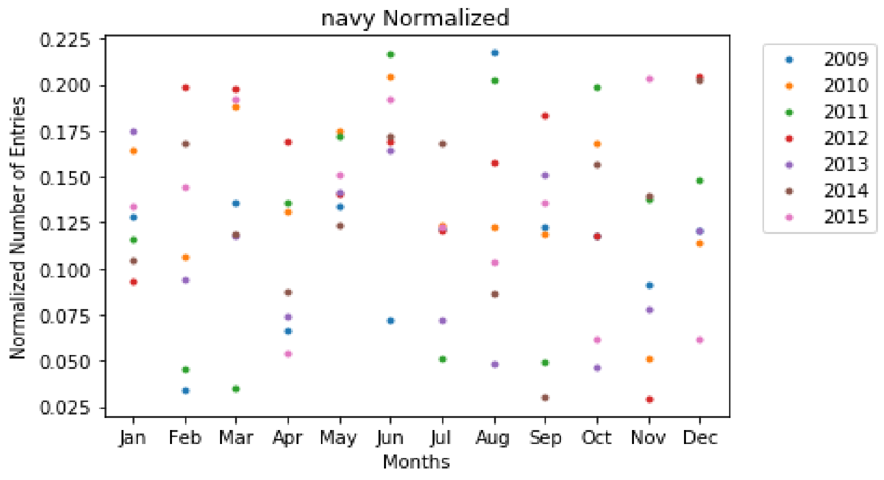


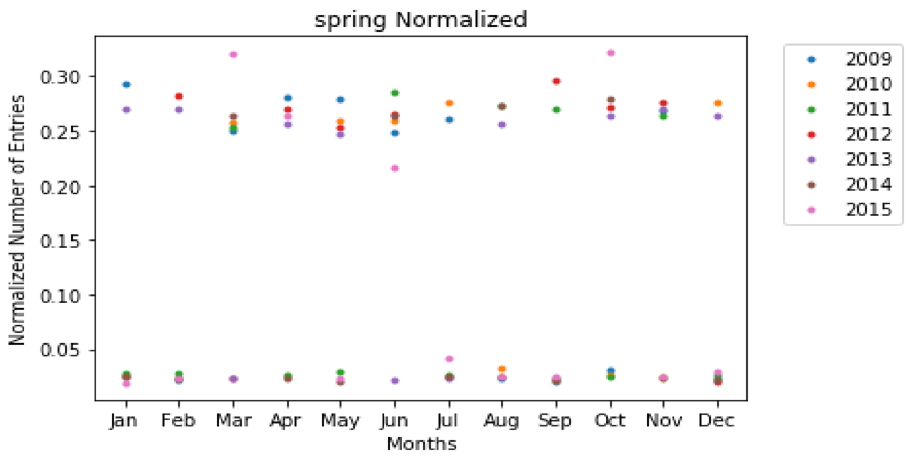
. Word Cloud of Tags

The next part of the exploratory data analysis was to see if the time data could be useful in the classifiers. The data covers eight years, from 2008 to 2015. Despite modelling the style and tag data in many ways, I was unable to find any trends in the data across years or any cyclic nature to the data by month. There are a few possible reasons for this. The first is that the data covers a large enough range of locations where seasonal differences would be washed out by the differences in climate at the difference locations. For example, users in Los Angeles would be likely to be able to wear shorts every month of the year, while users in New York City could only wear shorts June through August and users in Australia could only wear shorts in December through February.

When looking at the time data, an important step was to normalize all of the data so that I was graphing the percent of entries each month or year that contained a certain style or tag, not the number of entries. Graphing the total number of entries washed out any trends in favor of the much larger and more apparent trend of how many images were uploaded to the website in a given month or year. Overall, the data contains a rising number of submissions from 2008 to 2012 and a falling number of submissions from 2012 to 2015. Additionally, normalizing the data against the total number of entries per month did not reveal any trends in the top ten tags and styles.







Statistical Analysis

Statistical analysis for categorical data is tricky. I chose to perform a simple chi squared test to see if there was a statistically significant relationship between tags and styles that could potentially be exploited to build a model predicting styles using both tags and the images. The idea behind this model is that certain tags would correspond more closely to certain styles and that adding the tags as part of the training data on top of the images would improve the performance of classifying styles. I tested the null hypothesis that there is no statistically significant correlation between the tag “shorts” and the styles “summer” and “winter” and the null hypothesis that there is no statistically significant correlation between the tag “coat” and the styles “summer” and “winter”. There were 12638 occurances of the tag “shorts” and 12126 occurence of the tag “coat.” The percentages of each entry containing “summer” or “winter” also containing “shorts” or “coat” is shown in the table below.

	Summer	Winter
Shorts	7.25%	1.10%
Coat	1.30%	13.16%

Visually, this data looks to show the expected correlation. Tags of the summer associated “shorts” do in fact occur more on images that have been labeled as a summer style than a winter style. Tags of the winter associated “coat” also occur more on images that have been labeled with a winter style than a summer style. I completed this statistical analysis by performing a chi squared test to make sure this value was in fact statistically significant.

I was able to soundly reject both null hypotheses: summer and shorts are correlated more than winter and shorts with a chi squared value of 423 and winter and coats are correlated more than summer and coats with a chi squared value of 2126. As there is only one degree of freedom in each problem, these chi squared values correspond to p values of much less than 0.001, showing that each null hypothesis can be soundly rejected. Completing chi squared tests of this type for every potential pairing of tags and styles would take a long time, but this small investigation works as a proof of concept that there are meaningful relationships between the tags and the styles on the same images that could be used to classify the styles.

At the end of this process, the data is cleaned and ready to be used to classify the images. There was very little analysis or wrangling necessary for the images at this stage, as they all appear to be a very standard layout and format. Image manipulation will of course be a part of the next stage of the process as they will need to be broken down into matrix form, but given an initial visual inspection I do not foresee any difficulties with this process.

Initial Modelling

Once the data cleaning and visualization process was complete, I moved on to setting up initial models. The initial method I tried on the image classification problem was a deep neural network using convolution, batch normalization, and max pooling layers.

Preparing the images into pixel matrices for use in the network proved to be a challenge for my computer. I chose an image size of 150 by 100 pixels in order to keep the original scale. Operating on larger size pixel matrices than this also caused memory errors. At most, my computer was able to prepare 50,000 images, a little less than one fifth of the total images available for training. However, at most my computer was able to train on only 10,000 images. While 10,000 images was enough to show that there was promise in the idea, it was far from enough to create a deep learning network.

Once I began running sample trainings, I also increased the tag occurrences being used for the data in order to get more samples of each tag. I began with all tags with less than 100 mentions in the entire dataset being included. However, once I realized that I would only be able to train on 10,000 images, I raised this limit to 10000 mentions in the dataset. Therefore, the models are multilabel classification on 21 different tags. All tags were transformed into vectors using MultiLabelBinarizer from sklearn and the binarizer was saved to transform the results back into text. Another reason for cutting on a large number of tag occurrences was to reduce imbalance in the data set. Naturally there were some tags that were much more common than others. However, due to the same memory issues, it was difficult to resample the data set to create a balance training set. This certainly affected the precision of the network.

I began with a simple convolutional neural network setup in Keras, using two dimensional convolution layers, batch normalization, two dimensional max pooling, and dropout layers before the final dense layer for output. The final dense layer used a sigmoid activation, to facilitate multilabel classification. All other layers used rectified linear unit (ReLU) activation. Convolution was tested with symmetric windows between 2 and 6, with 4 performing the best as a window size. Batch size of approximately 256 and five epochs led to the best results, with any more epochs either being unnecessary as the loss and accuracy gain leveled off for the most part after 5 epochs. Four dropout layers were used to prevent overtraining. Two dimensional max pooling was used to reduce the dimensions between the layers. Different hidden layer structures were tested, with the most successful being five convolutional layers, each followed by batch normalization. Two dimensional max pooling and dropout were added after the first, third, and fifth batch normalizations. Finally, the model was flattened and two dense layers were applied to provide the output.

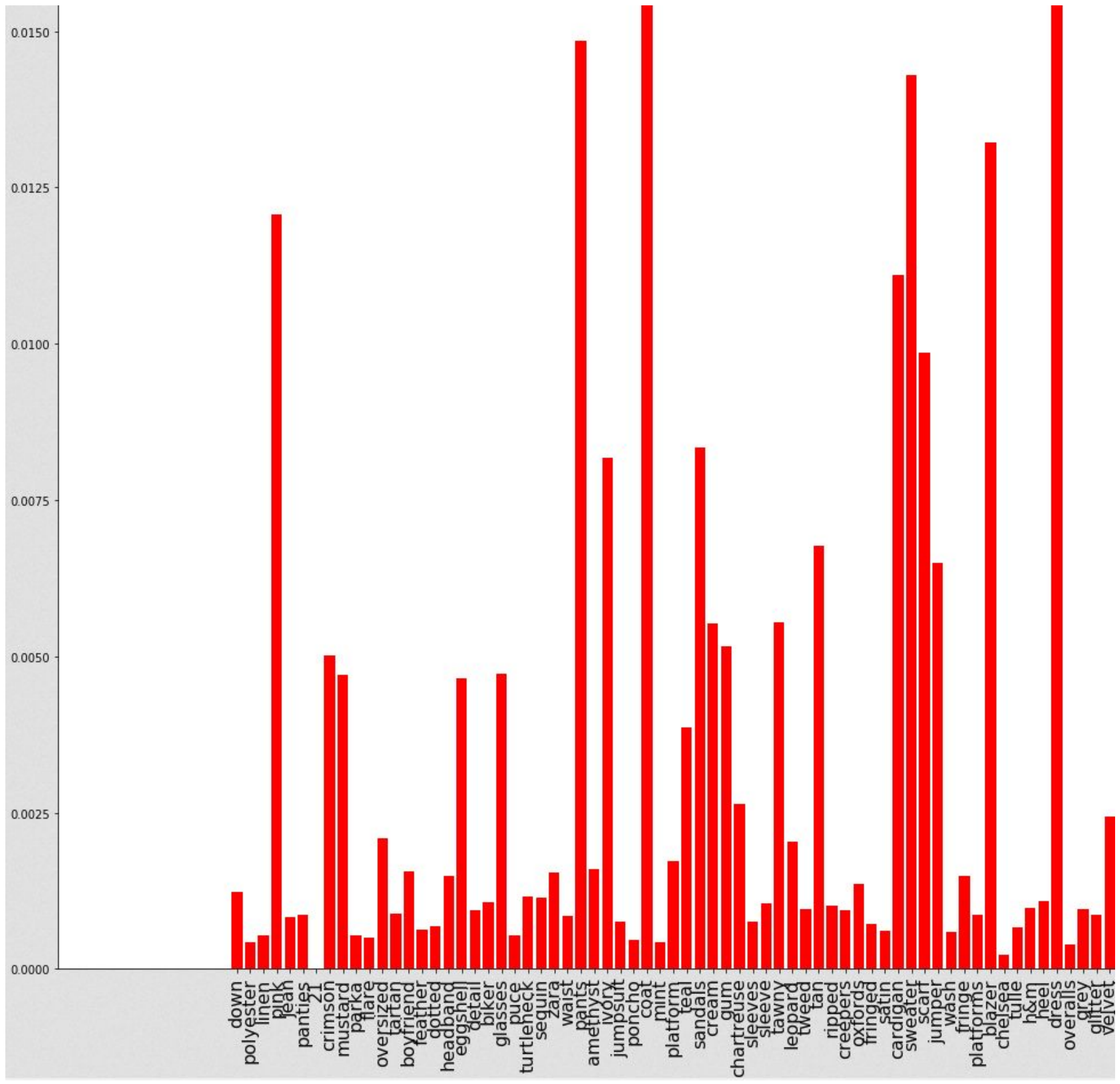
At this point in the analysis, the model has a high recall but low precision. With the classification parameter that a successful prediction is a value higher than 0.1 in the output matrix, most of the tags had a recall of close to 1 but a precision closer to 0.2. Therefore, the initial model was solving the classification by predicting almost every image to have every tag. This would potentially be solvable once more data points were added. Some optimization was performed on the model, but optimization was limited by the model runtime and knowing that more data points were needed to truly have a functional model. All of these issues precipitated the need to move the analysis onto a hosted virtual machine in order to train on the entire data set. Google Cloud virtual machines were tested to hold the analysis, however despite the much larger memory and better CPUs analysis ran prohibitively slow on the remote server. Therefore, due to time restraints, this method was abandoned.

Extended Analysis

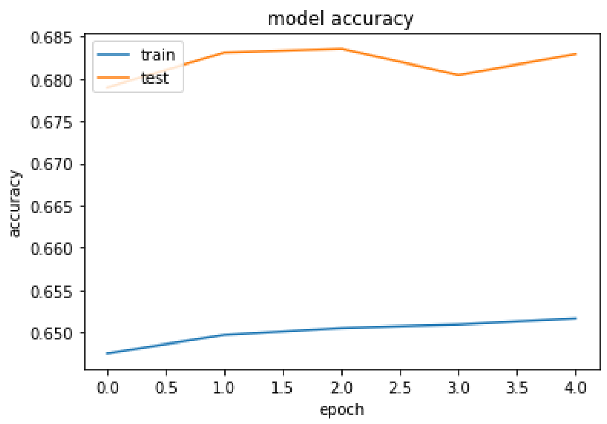
At this point, I turned my attention to the second idea for a model, predicting the styles using the tags on the images. While the optimal implementation of this model would also include information from the images themselves, this was not possible due to the same restrictions mentioned above. The full data set was able to be used for this part of the analysis. Tags were cut on having more than 100 occurrences throughout the data based on the idea that more data would be more helpful than trimming the variables to reduce redundancies and overtraining. After a brief analysis, I determined that the best course of action was to create a multiclass classifier to find the season of each outfit. Therefore, the final model was a four class classifier with outputs “winter”, “spring”, “summer”, and “fall” based on how each outfit was labeled by the user. I used a train/test split of 90% training, 10% testing to maximize the amount of data for training. Even with this split, there were still just slightly under 20,000 samples used for testing.

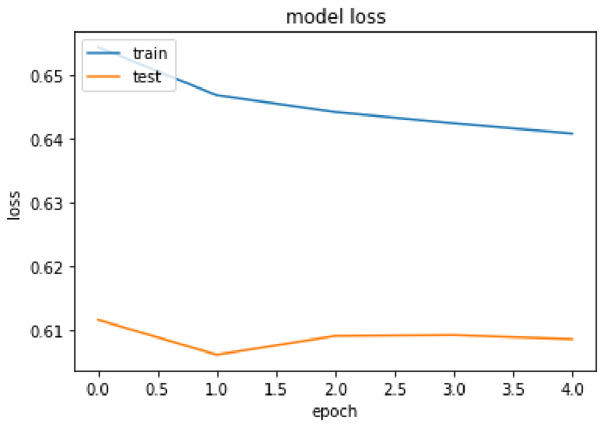
The training data was unbalanced, with approximately 90,000 samples being tagged as summer, 60,000 being tagged as spring and fall, and 30,000 being tagged as winter. Therefore, without resampling every model I tested would tend to over-classify summer. I used a basic resampling of winter, spring, and fall to bring all of the classes to similar frequency in the training data. In the end, this led to 390,037 samples to train the network on. I did not resample the testing data. After resampling, I used a multilabel binarizer to prepare the categorical data for the models.

The three methods I tested were PCA, random forest classifier, and a deep neural network. PCA was tried on 50,000 samples of data with three principal components. Graphical analysis of these three principal components did not lead to any separation of classes or insights into the data. The random forest classifier was the next model to be tried. While it was not very successful at classifying the seasons, with an average precision and recall of around 30%, it was useful to produce feature importances. Over a range of 263 tags, the feature importances aligned with what would be expected given the statistical analysis. Tags like “21”, “diy”, and “casual” that do not correspond to a physical feature of the outfit were given no feature importance. Tags like “strap”, “mesh”, and “polyester” which do correspond to physical features but not features that are strongly associated with any season had a small feature importance. The tags with the highest feature importances were either colors or tags like “coat”, “sweater”, and “shorts”, which have a strong association with whether. This analysis is interesting because it mostly conforms to the expected ideas about how a human would classify the season of an outfit. However, colors having such large feature importance was unexpected. Below is a small selection of the feature selection graph.

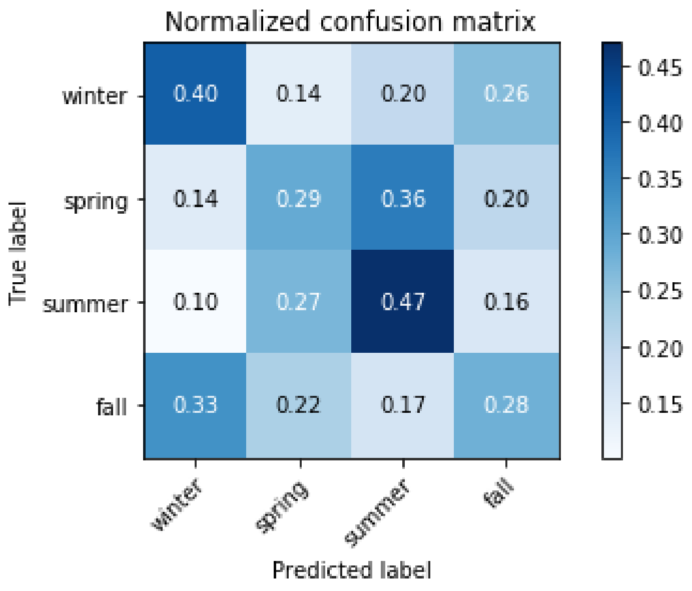


The most successful model was the deep neural network. While I tried multiple more complex layer structures, the most successful was a collection of simple densely connected feed forward network layers using ReLU activation and the Adam optimizer with an adaptive learning rate. The final dense layer used softmax activation for single classification. Both deep and wide structures were tested, having many hidden layers with few nodes or few hidden layers with many nodes respectively, and both had very similar results. Overall, with all architectures tested, the best test accuracy was approximately 68.5%, with a log-loss of approximately 28. Sample model loss and accuracy over five epochs is shown in the graphs below. For all models tested, the test accuracy is a few percent higher than the training accuracy. While this may seem counterintuitive, it comes from the fact that the model was best at classifying summer outfits. As discussed above, the training samples were regularized, but the test samples were not. Therefore, the model will do better overall on the test sample. There is a possibility that my method of resampling caused the model to perform less well on the other three seasons because it caused overtraining by showing the same samples multiple times. However, the model still performed better with this resampling than without it. Dropout layers were used to try to lessen this overfitting, and overall adding dropout layers with a dropout of 0.2 improved the performance of the model after five epochs.





The precision, recall, and confusion matrix for this model tell an interesting story. Spring and fall consistently had the lowest recall. This is seen in the confusion matrix with a high amount of crossover between spring and summer and winter and fall. It makes sense that these two pairs of seasons would have a large crossover, as people tend to start dressing in winter fashions in the fall and summer fashions in the spring. Summer had the highest precision and recall, possibily due unique items of clothing like sunglasses, shorts, and swimsuits and unique colors like white, which many in the fashion world consider a strictly summer color.



Season	Precision	Recall
Winter	0.33	0.40
Spring	0.37	0.28
Summer	0.65	0.47
Fall	0.42	0.28
AVG	0.48	0.36

Conclusions, Recommendations, and Future Work

There is clearly much that could be done to improve upon the concepts explored in this capstone. The first recommendation for future work would be to expand upon the image classification model and find an environment in which to explore and optimize that model. This is a robust dataset and the full potential of training a model on the images and metadata contained in it has not been reached yet.

There is much that could still be done to refine the seasons classification model. Using the feature importance found by the random forest classifier to create datasets using only the tags with high feature importance could give insight into how to expand this model for use in production level recommendation systems. By identifying only the most important tags for season classification, future data sets could be tagged by training a model to identify only those most important features. Therefore, it would be optimized for the best seasonal classifications using the least amount of time, effort, and chance for error in identifying the features of an outfit. In addition to creating a model using only the most important features, more sophisticated resampling methods should be employed to balance the training data to prevent the probable overfitting that reduced the effectiveness of the classification for winter, spring, and fall outfits.

Even though this model is not the most accurate in its current form, there is a lot of promise for business applications. Lumping spring/summer and fall/winter and doing a binary classification problem would improve the accuracy greatly, as the greatest confusion came between these two sets of seasons. From a business standpoint, this is still a valid option, as spring and summer fashions are frequently lumped together as warm weather fashions and similarly for fall and winter with cold weather.