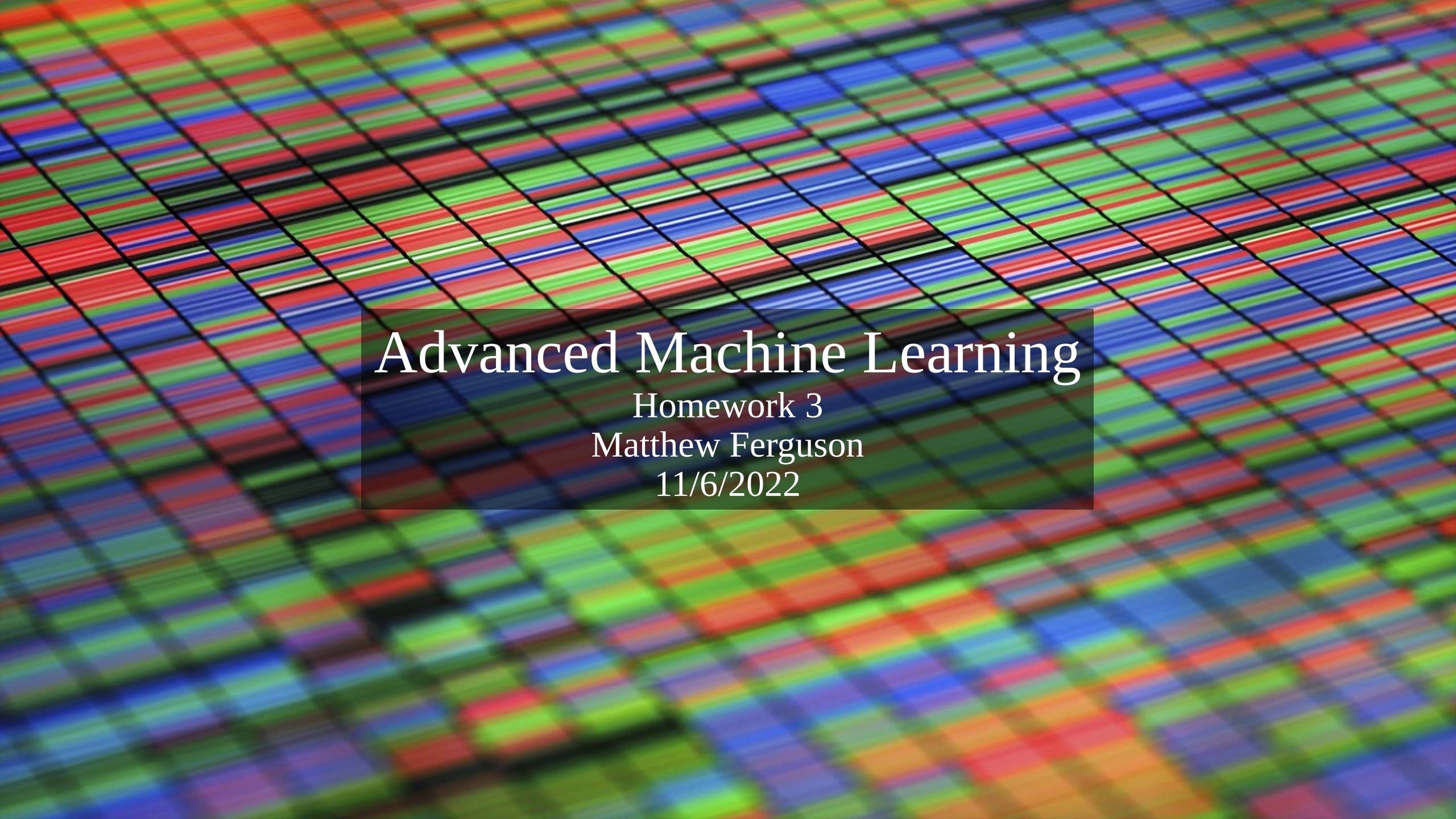

Advanced Machine Learning

Homework 3

Matthew Ferguson

11/6/2022

Requirements Digest

1. Create a paper discussing the ethical considerations of an “end-of-life advisor” machine learning model. Part 1 is attached separately.
   - Determine ethical decisions to be made
   - Specify the ethical grounds and form to be used
   - Address special ethical considerations (data, decision outcomes, system architecture, training/amount/process, monitoring, security/privacy)

2. Create a pair of logistic regression models to predict the binary variable isIncomplete in the National Football League play-by-play 2020 dataset.
   - Use only events where PlayType=Pass.
   - Prepare, normalize, split, drop features, and one hot encode the dataset appropriately.
   - Implement one model using all features.
   - Implement one model with feature selection.

Part 2 Code Overview

No preprocessing operation was performed on the dataset beyond reformatting it as a .csv file for efficiency reasons. We begin by importing the necessary libraries for our preprocessing and logistic regression tasks. Pandas, and NumPy are used for data manipulation. Sklearn is used for training machine learning models. Sklearn's model selection module will be used for training/test splits and the feature selection module will be used for sequential feature selection.

We load our play-by-play dataset as a data frame and drop the feature columns that are noted as to be deleted in the assignment brief. We filter the data frame down to passing plays. Then we use get dummies to return data frames of one hot encoded columns on the offense team, formation, play type (which always equal to pass), and pass type features. We drop the original categorical features, cardinality=1 features, and event data. Then we merge our one hot encoded features into our original data frame. We perform normalization on our preprocessed data frame and create a correlation matrix for later inspection.

We split our data frame into NumPy arrays containing the ground truth play complete indicator (y) and the preprocessed input features (x). We split our x/y using sklearn model selection into random sets of train and test data. 30% of records are used for testing, and the remainder for training. We first create a logistic regression model object, fit it using all preprocessed features, and then obtain its classification accuracy. Our logistic regression models have default settings except for the max iterations option which is set to 1000 to ensure convergence. Our second logistic regression model uses features obtained from sequential feature selection. Sequential feature selection used forward step feature selection and a tolerance of 0.001. The selected features are the only ones used for training model 2. Comparison of model performance is in the results section.

```
#AML HW 3
#Matthew Ferguson

import pandas as pd
import sklearn as sk
from sklearn import feature_selection, model_selection
import numpy as np

df=pd.read_csv(r'C:\Data\pbp-2020.csv')
df=df.drop(['DefenseTeam','IsChallenge','IsChallengeReversed','Challenger','IsMeasurement',
           'IsInterception','IsFumble','IsPenalty','IsTwoPointConversion','IsTwoPointConversionSuccessful',
           'RushDirection','YardLineFixed','YardLineDirection','IsPenaltyAccepted','PenaltyTeam','IsNoPlay',
           'PenaltyType','PenaltyYards', 'Description'], 1)
df=df.loc[df['PlayType']=='PASS']

d1=pd.get_dummies(df['OffenseTeam']).astype(int)
d2=pd.get_dummies(df['Formation']).astype(int)
d3=pd.get_dummies(df['PlayType']).astype(int)
d4=pd.get_dummies(df['PassType']).astype(int)

df=pd.concat([df,d1,d2,d3,d4],1)
df=df.drop(['OffenseTeam','Formation','PlayType','PassType'], 1)
df=df.drop(['NextScore','TeamWin','SeasonYear','IsRush','IsPass','IsSack','PASS','GameDate','GameId'], 1)

df=(df-df.min())/(df.max()-df.min())
cm=df.corr()

y=np.array(df['IsIncomplete'])
x=np.array(df.drop(['IsIncomplete'], 1))

(x_train, x_test, y_train, y_test) = model_selection.train_test_split(x, y, test_size=0.3, random_state=2)
model_1=sk.linear_model.LogisticRegression(max_iter=1000)
model_1.fit(x_train,y_train)
m1_acc=model_1.score(x_test,y_test)

model_2=sk.linear_model.LogisticRegression(max_iter=1000)
sfs=feature_selection.SequentialFeatureSelector(estimator=model_2,n_features_to_select='auto'
                                                ,tol=float(0.0001), direction='forward')
sfs.fit(x,y)
feature_bool=sfs.get_support()
x_new=sfs.transform(x)
(x_train_2, x_test_2, y_train_2, y_test_2) = model_selection.train_test_split(x_new, y, test_size=0.3, random_state=2)
model_2.fit(x_train_2,y_train_2)
m2_acc=model_2.score(x_test_2,y_test_2)
```

Part 2 Results

We created a correlation matrix and included only the correlations that were statistically significant (>0.05) to our target, incomplete pass. Note that we had 56 features after preprocessing and that only 9 had a correlation greater than 0.05. Most features are not useful at predicting if a pass was completed and this indicates feature selection will be a highly useful step in optimizing our logistic regression model. Model 1, the model trained on all features, has a test accuracy of 92.1% which is a strong result. Model 2, the model with a feature selection step, outperformed model 1 with an overall test accuracy of 94.6%. This is an interesting result as our second model can make more accurate predictions with less data.

Sequential feature selection chose 3 features when set to forward step and a tolerance of 0.001. We experimented with higher tolerances and found it would result in selection of only 1 or 2 features. The selected features were Yards, Touchdown, and Deep Left. It is interesting that feature selection resulted in a basket of features which was different than the highest absolute correlation features. It implies that there is more useful information gained from features across the range. Yards was most correlated with Series First Down(0.63) which is likely the underlying reason why both features were not selected despite them both having the highest absolute correlation. Deep Left and Deep Right were both one hot encoded from our pass type feature, and both were highly correlated with our target but not each other like the previous features. If tolerance was lowered further, I wonder if Deep Right would be picked as an additional feature.

Regarding Model 1 and 2 accuracy, my intuitive belief would be that more features would ‘aid’ model accuracy in that the more statistically insignificant features are added, the more diminishing returns are gained with respect to accuracy. That is to say that you might have a model that is 99% as good as another with $1/10^{th}$ the features. This intuitive belief rejected the idea that you could have a model trained using the same architecture and dataset as another and achieve superior accuracy with a subset of that model’s features. However, this is not the case, we trained on a subset of features using the same parameters and architecture and achieved superior accuracy. Our result does help to prove the phrase ‘the simplest possible model is generally best’.

Table 1. Model Results

	Model 1	**Model 2**
Features	56	3
Accuracy	92.1%	94.6%

Table 2. Significant Target Correlations

	IsIncomplete
Deep Left	0.1400
Deep Right	0.1322
Yard Line	0.0700
Down	0.0513
Short Middle	-0.0537
Short Left	-0.0874
Touchdown	-0.1606
Series First Down	-0.5132
Yards	-0.5346