Football Manager Dataset

BFOR 516 Final Project

Nicholas Lopez, Sajjad Khan, Jackson Nahom, Omer Keskin

https://github.com/nahomjd/BFOR-516-Project

Fall 2021

Submitted for Dr. Lee Spitzley

Executive Summary

This project includes two parts, (i) predicting performance metrics for football players, and (ii) recognizing digits using machine learning algorithms.

The first part of the project addresses three specific problems about prediction of various parameters of football (i.e., soccer) players. Football clubs pay attention to predicting performance of players for various purposes, such as training and transfers. We used the dataset of Football Manager 2020 game for the project. The dataset includes various parameters for almost 150,000 players.

The first question we asked was can we predict a player's Current Ability? In order to predict this, we used six regression models to test. We discovered that our MLP regression model was the best for predicting a player's Current Ability. It performed accurately and featured an RMSE of 4.5679, an MSE of 20.8655, and an R-squared of 0.9811. It also had the lowest amount of error in predictions (average predicted difference from the actual value) out of the six regression models we tested at 3.4242. The second question we asked was can we predict a player's Potential Ability? Like the previous question we used the same six regression models to try and answer the question. We determined that the Random Forest and MLP regression models performed the best as they featured the lowest RMSE in predictions with 11.5368 and 10.4145, respectively. These models were not nearly as accurate as our Current Ability prediction models. Finally, came the classification question can we predict which players are in a Top 10 league? Seven different classification models were used to try and answer the question. At best, our best model performed averagely. The accuracy mostly stems from predicting a player who is not in a top 10 league. However, the intent was to predict if they were in one of those leagues and for that case, we were around a coin flip odds. Overall, we are content with the outcomes of predicting potential and current ability from the regression models.

The second part of the project focuses on recognizing handwritten digits. Predicting handwritten values is pretty important in some businesses and most technologies like OCR use it. We used a dataset digit-recognizer from Kaggle that consists of thousands of handwritten images converted into a CSV dataset.

The question we asked was can we predict handwritten digits? To solve this classification problem, we developed three different classification models. These classification models performed relatively well. All models successfully predicted digit values. The Random Forest model was best at predicting based on evaluation statistics. It predicted digit values with an accuracy of 0.9998. This accuracy level gave us confidence that we could solve the question with our model

Part I – Predicting Football Player Abilities

Problem & Data Descriptions

Part I of the project addresses problems regarding predicting various parameters of football (i.e., soccer) players. It is deemed important to predict the overall performance of a player for managers and directors of football clubs for various purposes, including training and transfers. We used secondary data to address this generic problem.

Organization Problem Statement

We have identified the following Business Questions for the purpose of this project:

1. *Can we predict a player's Current Ability?*
2. *Can we predict a player's Potential Ability?*
3. *Can we predict which players are in a Top 10 league?*

Dataset

We used Football Manager 2020 Dataset for this project. This dataset provides a collection of football players and their statistics, such as age, position, club, nationality, value, wage, all player attribute components. The size of the CSV file data is 35MB that consists of 64 Columns and 144750 rows, each represent one specific player identified by their name.

Acquisition and Relevance

Dataset is acquired from Kaggle website. The original source of the dataset is the Football Manager 2020 game.

Realness

Football Manager is a simulation game that could best be described as a spreadsheet simulator. The developer studio of the game developed this dataset to use for the game. The data used comes from a network of 13,000 scouts of at varying levels that send reports to the developer Sports Interactive.

The data is so expansive and detailed that some high-level clubs pay Sports Interactive for access to their database. For example, Everton publicly signed a database deal with Sports Interactive back in 2008 in this manner.

Limitations

Like other sports data, this dataset is likely to be incomplete to some degree. There will be errors/omissions and it will not encompass each goal or kick every player has made.

This data also does not include statistics about players' likelihood of injuries. This likely would have had a tangible effect on players' long-term Potential Ability.

Data Exploration

List of Relevant Variables

Key columns include but are not limited to Wage, Value, Age, Long Shots, Long Throws, Passing, Dribbling, Aggression, and Stamina. 56 out of 64 columns were used for analyses.

Sample data for relevant variable

Table 1 provides a sample of the dataset. Some of the important variables are included in the table: the position he or she plays, the club the player belongs to, the division (league) the club is in, Current Ability (CA), Potential Ability (PA), Age, best position of the player, best role, total value, wage, and some other parameters, such as passing, dribble, long throw, balance, agility, acceleration, and nation.

Table 1. Sample Data for five players with some of the important variables (transposed)

Name	Lionel Messi	Cristiano Ronaldo	Neymar	Zlatan Ibrahimovic	Mohamed Salah
Position	AM (RC), ST (C)	AM (RL), ST(C)	M(L), AM(LC), ST(C)	ST (C)	AM (RL), ST (C)
Club	Barcelona	Juventus	Paris SG	Milan	Liverpool
Division	Spanish 1st Div.	Italian Serie A	Ligue 1	Italian Serie A	English Premier
CA	195	195	186	145	179
PA	200	196	190	185	184
Age	32	34	27	37	27
Best Pos	AM (R)	ST (C)	AM (L)	ST (C)	AM (R)
Best Role	IF	CF	IW	DLF	IF
Value	73,000,000	31,000,000	91,000,000	5,250,000	79,000,000
Wage	1,200,000	950,000	600,000	180,000	200,000
Passing	20	15	16	15	14
Dribble	20	15	20	11	16
Long Throw	4	3	2	8	4
Balance	19	14	14	19	17
Agility	19	13	18	12	16
Acc	18	15	17	6	18
Nation	ARG	POR	BRA	SWE	EGY
48 more fields	…	…	…	…	…

Number of observations

144,750 rows (players) a 75/25 train/test split was used for the CA and PA questions, while a 60/40 train/test split was used for our classification question.

Preprocessing steps

Turned Wage and Value transformed into log values. Turned Nationality, preferred foot, best position, and best role into dummy columns for each unique value. We gave a classification of 1/0 for a player in the top 10 league.

Variable distributions

Figure 1 presents the histogram for current ability values for each player. It can have a value from 0 to 200. There are negligible amount of players who has a CA more than 165.

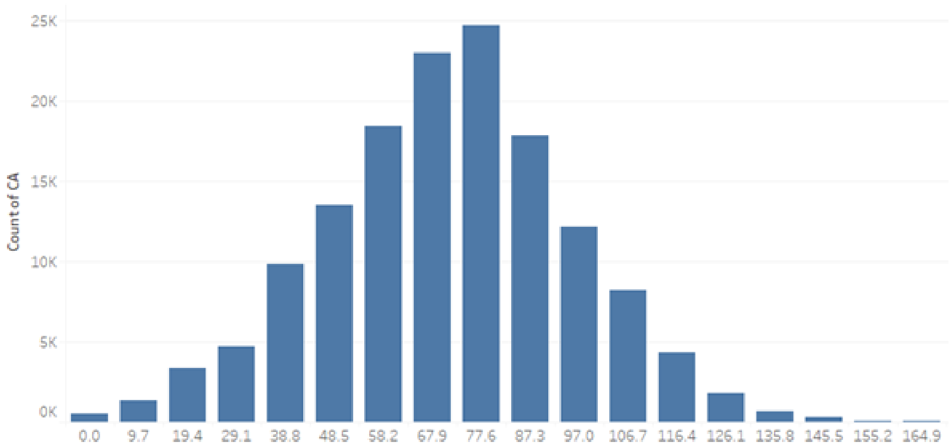


Figure 1. Histogram for Current Ability (CA)

Figure 2 presents a histogram for the potential ability (PA) of the players. Compared to CA, the values are slightly pushed to right since some players have a higher potential ability than their current ability.

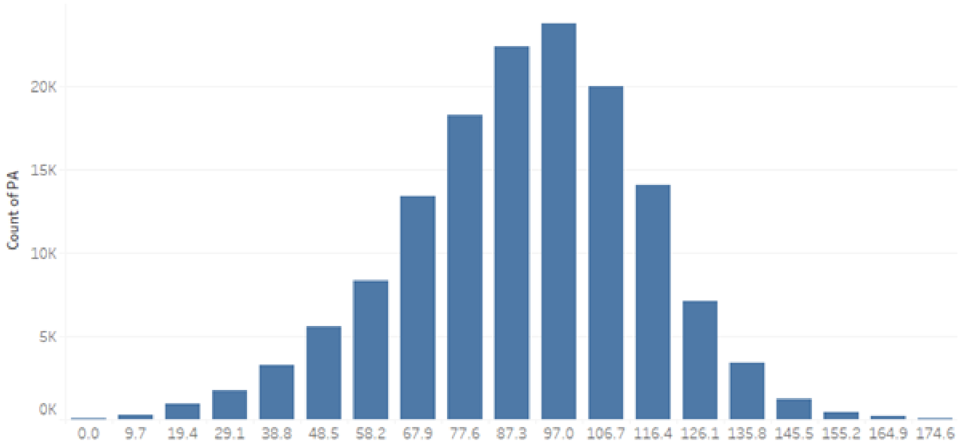
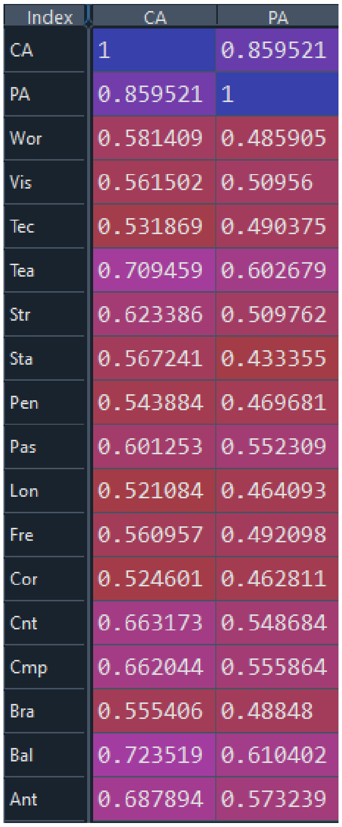


Figure 2. Histogram for Potential Ability (PA)

Interesting correlations

A correlation matrix was calculated for all columns using Python; however, since there are many columns, it is not practical to include in the report. We produced another correlation matrix that focuses only on CA and PA where highly correlated fields with these two were highlighted as can be seen in Figure 3.



Index	CA	PA
CA	1	0.859521
PA	0.859521	1
Wor	0.581409	0.485905
Vis	0.561502	0.50956
Tec	0.531869	0.490375
Tea	0.709459	0.602679
Str	0.623386	0.509762
Sta	0.567241	0.433355
Pen	0.543884	0.469681
Pas	0.601253	0.552309
Lon	0.521084	0.464093
Fre	0.560957	0.492098
Cor	0.524601	0.462811
Cnt	0.663173	0.548684
Cmp	0.662044	0.555864
Bra	0.555406	0.48848
Bal	0.723519	0.610402
Ant	0.687894	0.573239

Figure 3. Correlation Matrix for important columns (Current Ability and Potential Ability)

Informative visualizations

We used Seaborn library of Python to visualize distribution of some of the variables. Figure 4 presents how CA and PA are related for each player. As can be seen from this plot, many players have a higher potential than their current ability. Moreover, the color coding of the plot represents the value of the players where light-colored data points have up to 2 million dollars value and darkest data points indicate the players who have approximately 8 million dollars value. As expected, higher current and potential ability correlates with the high value of the player.

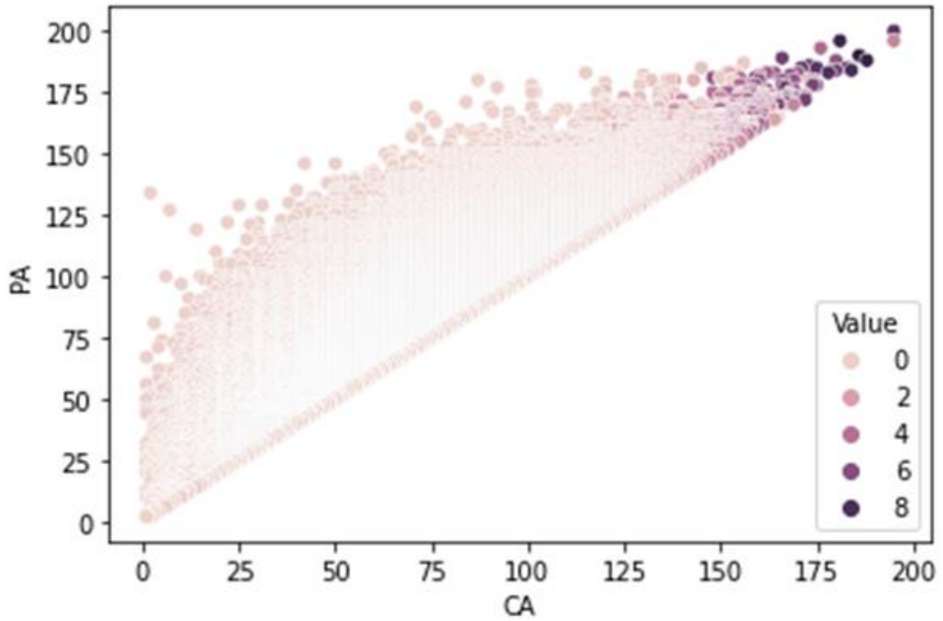


Figure 4. Current Ability, Potential Ability, and Value

Figure 5 represents Value of the players versus their wages. Also, Current Ability indicates the darkness of the data points. This plot also indicate that low CA value results in lower wage and value.

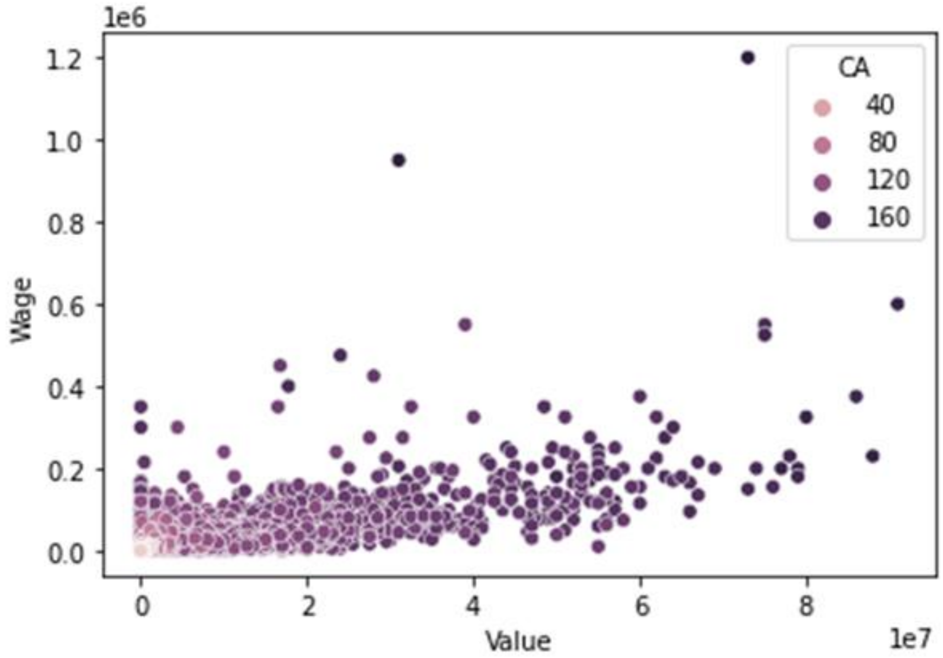


Figure 5. Value, Wage, and Current Ability

Figure 6 visualizes the origins of players. Most players are coming from Argentina, Brazil, and European countries. This map is generated using Tableau software.

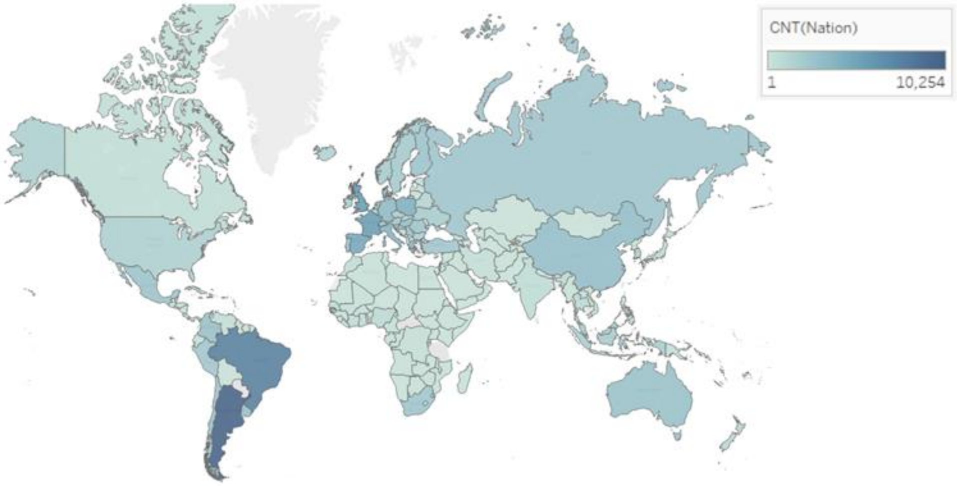


Figure 6. Number of Players per Country

General conclusions on dataset

The dataset has a lot of useful and interesting columns for real football players. It is not only insightful, but also fun to explore and run machine learning models to predict trends and various parameters.

Regression Questions

Regression Question 1: Predicting Current Ability

Question 1: Can we predict a player's Current Ability (CA)?

Regression is an appropriate method for this question since CA is a continuous variable between 0-200 and predictor variables are mostly numeric. The predictors that were not numeric were nationality, preferred foot, best position, and best role. These non-numeric variables were converted to dummy variables to be used for the regressor models.

We used 6 regression Models to fit and test to compare and find the best model to predict CA. The models we used are MLP, Random Forest, K Nearest Neighbors, Decision Tree, and SVR. After converting our non-numeric variable to dummy columns we ended up with 331 independent predictor variables.

Evaluation Statistics

Table 2 presents the summary statistics of the performance of the models we developed to address the first business question.

Table 2. Summary statistics of the model performance for Question 1

Regression Models	RMSE	MSE	R-Squared	Avg Predicted Difference from Actual
MLP	4.5679	20.8655	0.9711	3.4242
Random Forest	6.5051	42.3160	0.9905	4.8273
KNeighbors	7.9607	63.3737	0.9316	6.1693
Decision Tree	10.3935	108.0259	1.0	7.7088
SVR	7.3265	53.6781	0.9141	5.6956
OLS	6.3130	39.8539	0.994	4.8560

The Three best models came out to be the Random Forest, MLP, OLS models. Since what we are trying to do is correctly predict a player's Current ability from 0 to 200 the evaluation statistic that is the most meaningful due to it telling us about the error in predictions. For this reason, we can conclude our MLP model is the best for predicting a player's current ability reasonably accurately.

Regression Question 2: Predicting Potential Ability

Question 2: Can we predict a player's Potential Ability (PA)?

Regression is an appropriate method for this question since PA is a continuous variable between 0-200 and predictor variables are mostly numeric. The predictors that were not numeric were nationality, preferred foot, best position, and best role. These non-numeric variables were converted to dummy variables to be used for the regressor models.

We used 6 regression Models to fit and test in order to compare and find the best model to predict PA. The models we used are MLP, Random Forest, K Nearest Neighbors, Decision Tree, and SVR. After converting our non-numeric variable to dummy columns we ended up with 331 independent predictor variables.

Evaluation Statistics

Table 3 presents the summary statistics of the performance of the models we developed to address the first business question.

Table 3. Summary statistics of the model performance for Question 2

Regression Models	RMSE	MSE	R-Squared	Avg Predicted Difference from Actual
MLP	10.4145	133.0992	0.8485	7.9796
Random Forest	11.5368	108.4627	0.9705	8.7813
KNeighbors	13.4783	181.6657	0.8086	10.4063
Decision Tree	16.8340	283.3849	1.0	12.7955
SVR	13.4266	180.2736	0.7148	10.3696
OLS	12.3311	152.0564	0.984	9.5452

The two best models turned out to be the Random Forest and MLP models. Since what we are trying to do is correctly predict a player's Potential Ability from 0 to 200 the evaluation statistic that is the most meaningful due to it telling us about the error in predictions. For this reason, we can conclude our MLP model is the best for predicting a player's Potential Ability. Although we are not as accurate as the CA models so it would be better suited to use this model as predicting the range a player's potential is ±10.4145 of the predicted value.

Classification Questions

Classification Question: Predict Whether in Top 10 League

Question 3: Can we predict which players are in a Top 10 league?

To solve this question, we needed to first decide what the top 10 leagues were. We used this list from this article[1]. The leagues that the article concluded were the top 10 in the world were: Spanish First Division, Italian Serie A, Ligue 1 Conforama, Bundesliga, English Premier Division, Argentine Premier Division, Brazilian National First Division, Eredivisie, Turkish Super League, and Portuguese Premier League. We gave every player that is on a team in this league a 1 and those who were not a 0. This is a player in a top 10 league is our dependent variable we are trying to predict. There were 10,724 players in those 10 leagues and 134,026 that were not in those 10 leagues. This makes our data set lopsided. We predicted which players are in a Top 10 League by using seven classification models: Decision Tree, Random Forrest, MLP, Guassian NB, Ada Boost, Logistical Regression, and Voting Classification. The Voting Classification uses Decision Tree, Random Forrest, MLP, and Ada Boost. Classification was used because we are predicting a binary variable that is either 1, or 0. We used a 60/40 train/test split for these models. We used 56 independent variables as prediction variables for predicting our dependent variable. Table 4 presents the Evaluation statistics that we used to compare our models.

Table 4. Summary statistics of the model performance for Question 3

Models	AUC	Log Loss	Accuracy	Macro Accuracy
Decision Tree Classifier	0.906	0.4095	0.95	0.78
Random Forrest Classifier	0.944	0.1441	0.95	0.73
MLP Classifier	0.942	0.1389	0.95	0.77
Voting Classifier	0.954	0.1928	0.95	0.77
Gaussian NB	0.755	1.1836	0.77	0.56
Ada boost Classifier	0.939	0.6496	0.94	0.71
Logistic Regression	0.887	0.1834	0.93	0.62

[1] https://soccermodo.com/best-soccer-leagues-in-the-world/

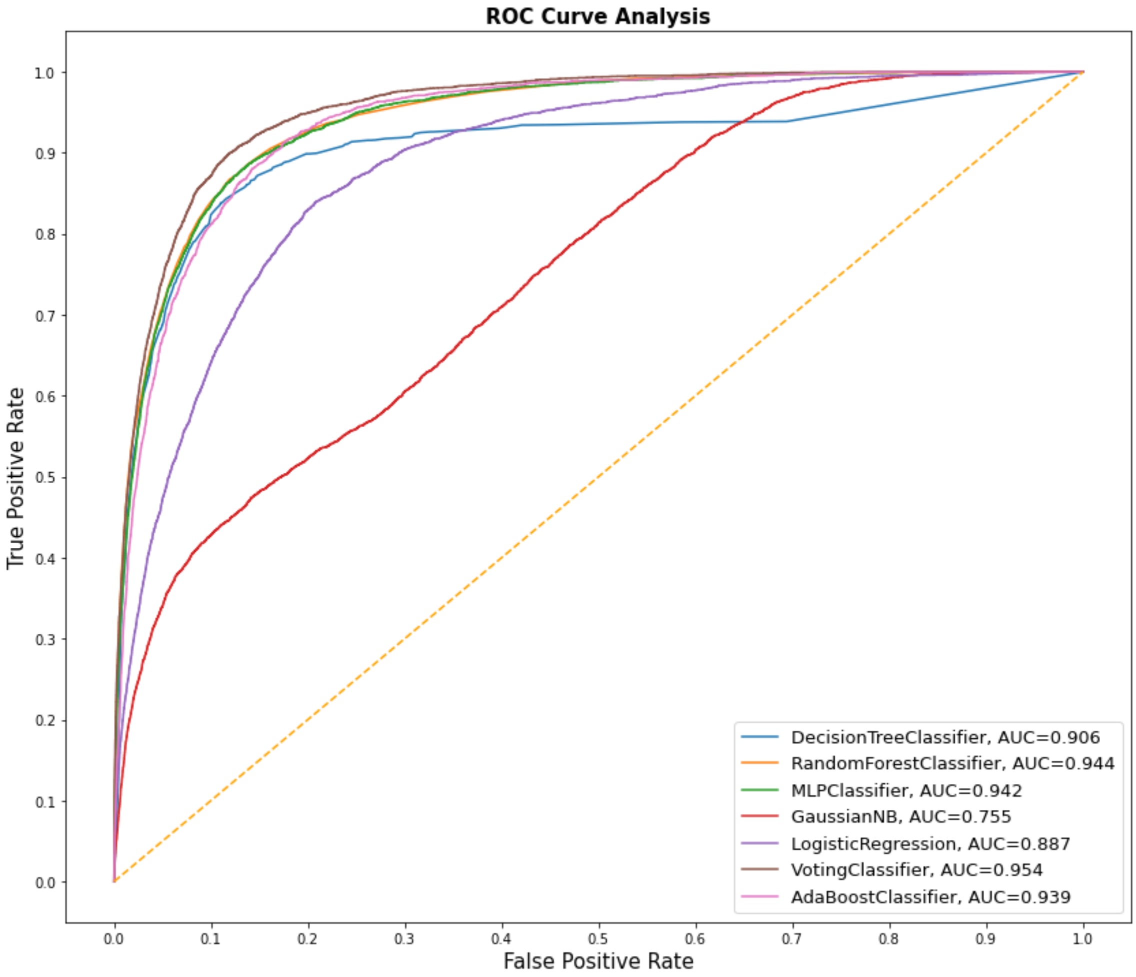


Figure 7. AUC curves for the models

According to the AUC curves (Figure 7) the voting classifier performs best in terms of positives against false positives. Since our test data set is lopsided, the Accuracy statistic is not the best evaluation to use, instead in this case the Macro Average is a better guide to which model performed best. Using our evaluations, we can conclude the best performing models are the Decision Tree, MLP, Voting, and Random Forrest models. To get a better look at the four we investigated each model's classification matrix.

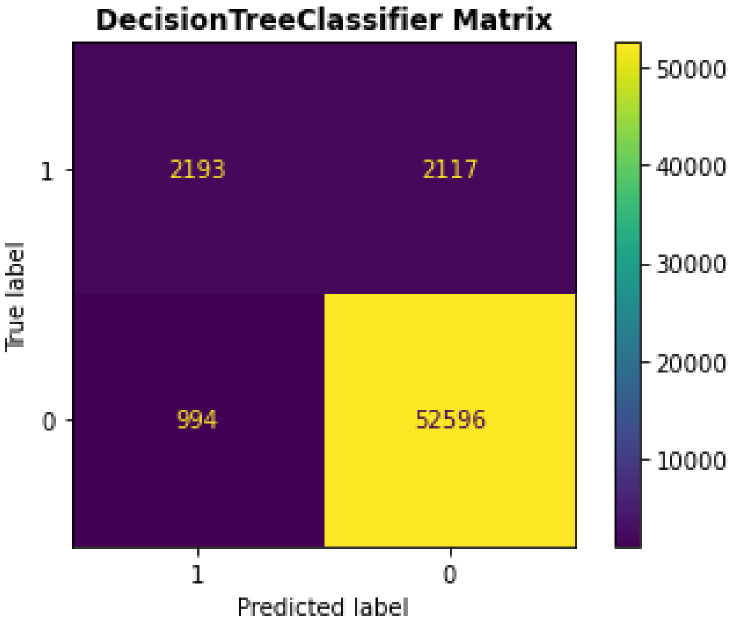


Figure 8. Decision Tree Classification Matrix

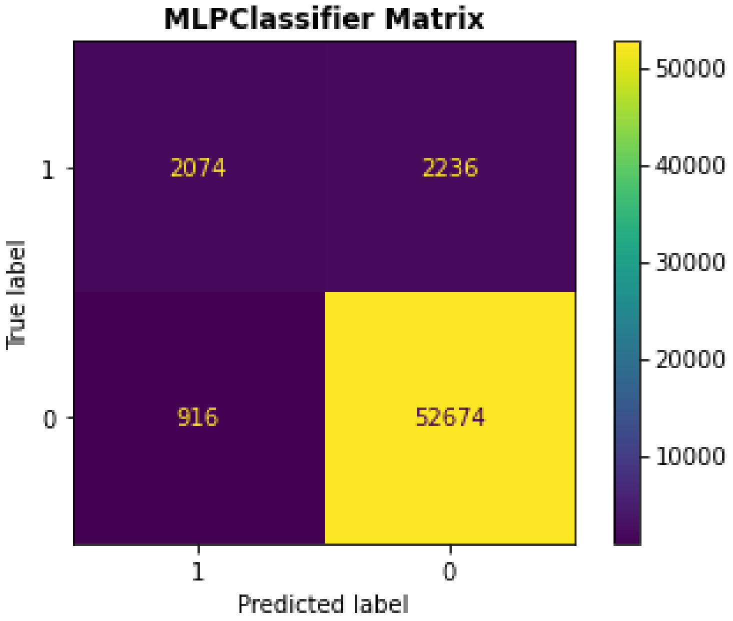


Figure 9. MLP Classification Matrix

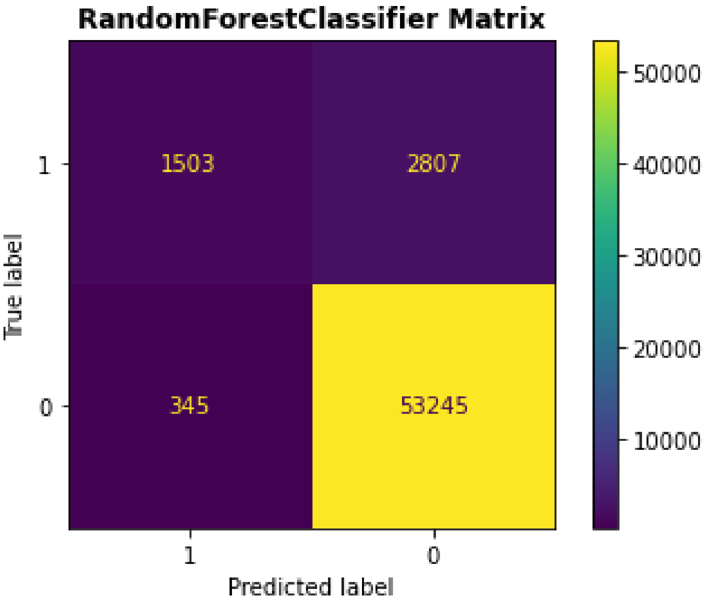


Figure 10. Random Forrest Classification Matrix

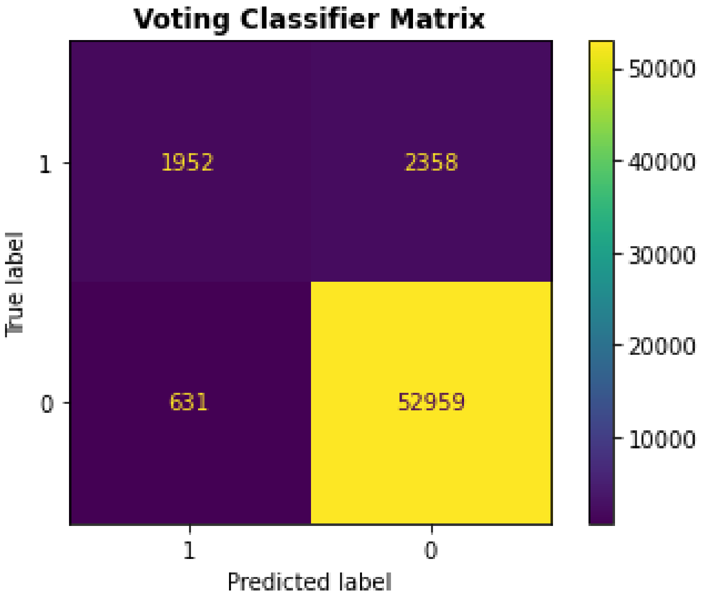


Figure 11. Random Forrest Classification Matrix

Looking at the classification matrices we can conclude the best performing models are the Decision Tree and MLP models. This is because the goal of the models is to predict if a player is in a top 10 league. For that case, the best performing model is the Decision Tree as the only model predicting the majority of players in a top 10 league, even though it only performed best via Macro Average. In the end we were not able to vary accurately predict the question we asked. Our best model was only able to predict a little over 50% of the players in a top 10 league.

Conclusion and Future Directions

We can accurately predict a player’s current ability ±2.3%. However, predicting a player’s potential is not extremely accurate. Our player potential model would be better suited used as a range model where the prediction is ±10.4145 of the predictive value. Our classification model predicts if a player is in a top 10 league needs some improvement if it were to be used as intended. The accuracy mostly comes from predicting a player is not in a top 10 league, however this is not the model's purpose. We were only able to predict about half of the players in a top 10 league of our test data set. Overall, we believe the regression models performed well for their intended purposes, while our classification fell flat on its face at best.

For future work, we can try a different approach regarding predicting if a player is in a top 10 league. It would take more than tuning to make our current model useful in its predictions.

Part II – Recognizing Digits

Problem & Data Descriptions

We have identified the following Business Question for the purpose of Part II of the project:

Can we predict a handwritten digit?

Dataset

We used the digit-recognizer Dataset for this project. The data contains gray-scale images of hand-drawn digits, from zero through nine. Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker.

The size of our data set is 128 MB. It has 785 Columns, and 42,000 Rows (number of handwritten numbers).

Acquisition and Relevance

Dataset is obtained from Kaggle website. The dataset is based on handwritten digits.

Realness

The Dataset is based on handwritten digits one through nine. Each image was transformed into a grey scaled image and then this 28x28 image was converted into 784 pixels to form a table.

Limitations

The dataset has two parts train and test. The test dataset does not contain actual values. Running our model on the test dataset will perform the predictions but we will not be able to evaluate the results.

The current dataset is in table form and already pre-processed images. Prediction using images is not supported and it will require additional work.

Data Exploration

List of Relevant Variables

The dataset consists of 785 columns where the first column is “label”, the actual value of the digit, and the rest of the column represents all pixels of the image.

So, all the columns are important for model building.

Preprocessing steps

Each column has 784 columns. The first column is the actual value of the digit, the remaining 784 columns represent each pixel of the image. In order to process and build a model based on the values converted all columns for a row into a 28x28 matrix/2D array.

Occurrence of Digits

Figure 12 shows the frequency of each digit in the dataset. Digit 1 appeared the most.

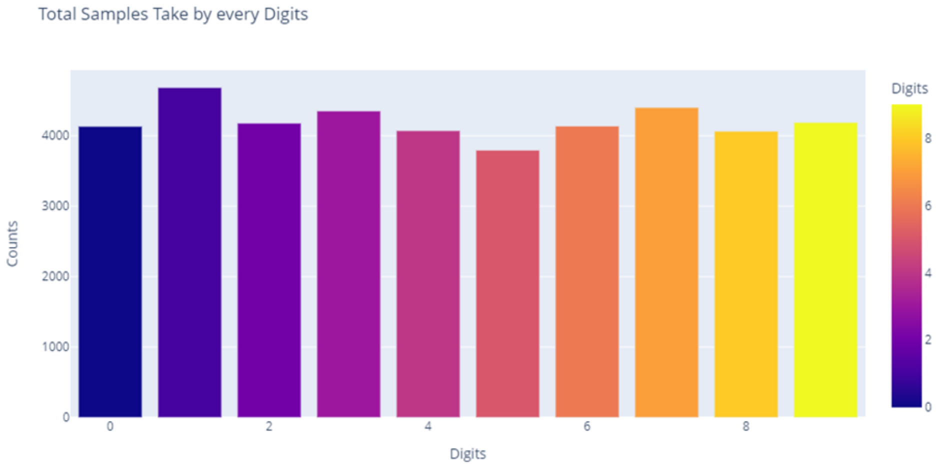


Figure 12. Frequency of each digit in the dataset

Visuals of a Digit

Figure 13 shows the visualization of a sample handwritten digit.

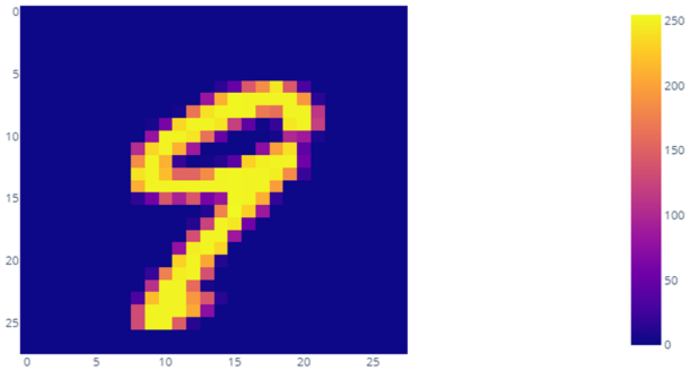


Figure 13. A sample of handwritten digit

Classification Question

Can we predict a handwritten digit from images?

Classification is an appropriate method for this question. We used three classification models to fit and test to compare and find the best model to predict the digit. The models we used are MLP, Random Forest, and Support Vector Machine (SVM).

All models performed relatively well and However Random Forest was the best model to recognize the digits. The AUC value of random Forest is 0.99982. Evaluations of all models are listed in the evaluation section.

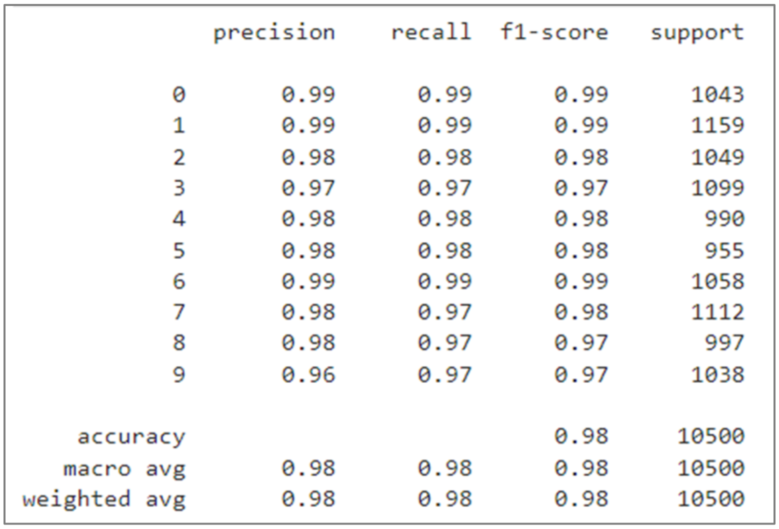
Evaluation Statistics

We used Average Accuracy, Average Precision, F1-Score, and Average AUC to evaluate our models.

Table 5. Summary statistics of the model performance for Question 2

Model	Avg. Accuracy	Avg. Precision	Avg. F1-Score	Avg. AUC
SVM	0.98	0.98	0.98	0.99961
Random Forest	0.99	0.99	0.99	0.99982
MLP	0.97	0.97	0.97	0.99875

SVM statistics are provided in Figure 14.



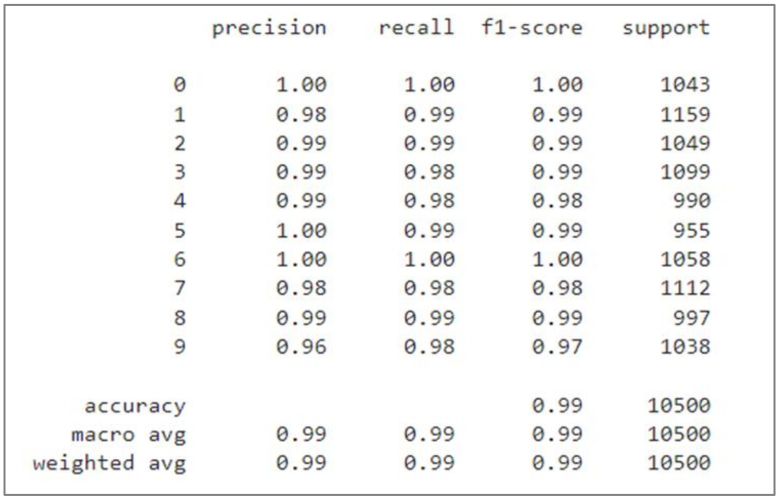
```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1043
           1       0.99      0.99      0.99      1159
           2       0.98      0.98      0.98      1049
           3       0.97      0.97      0.97      1099
           4       0.98      0.98      0.98       990
           5       0.98      0.98      0.98       955
           6       0.99      0.99      0.99      1058
           7       0.98      0.97      0.98      1112
           8       0.98      0.97      0.97       997
           9       0.96      0.97      0.97      1038

    accuracy                           0.98     10500
   macro avg       0.98      0.98      0.98     10500
weighted avg       0.98      0.98      0.98     10500
```

Figure 14. SVM statistics

Random Forest statistics are provided in Figure 15.



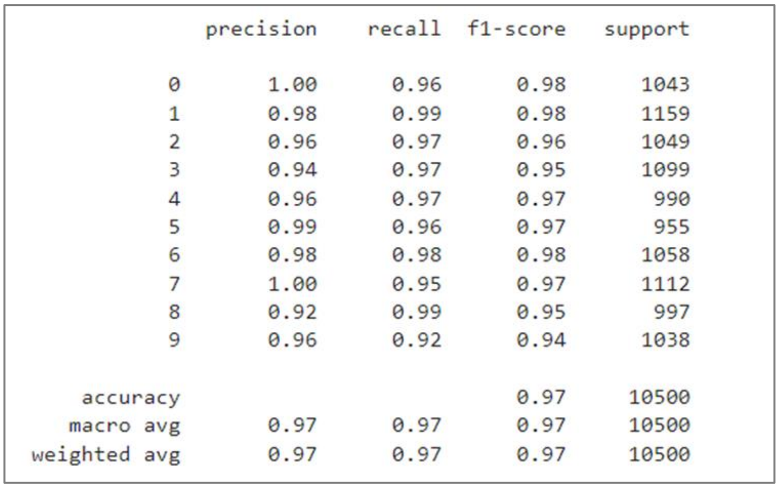
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1043
           1       0.98      0.99      0.99      1159
           2       0.99      0.99      0.99      1049
           3       0.99      0.98      0.99      1099
           4       0.99      0.98      0.98       990
           5       1.00      0.99      0.99       955
           6       1.00      1.00      1.00      1058
           7       0.98      0.98      0.98      1112
           8       0.99      0.99      0.99       997
           9       0.96      0.98      0.97      1038

    accuracy                           0.99     10500
   macro avg       0.99      0.99      0.99     10500
weighted avg       0.99      0.99      0.99     10500
```

Figure 15. Random forest statistics

MLP statistics are provided in Figure 16.



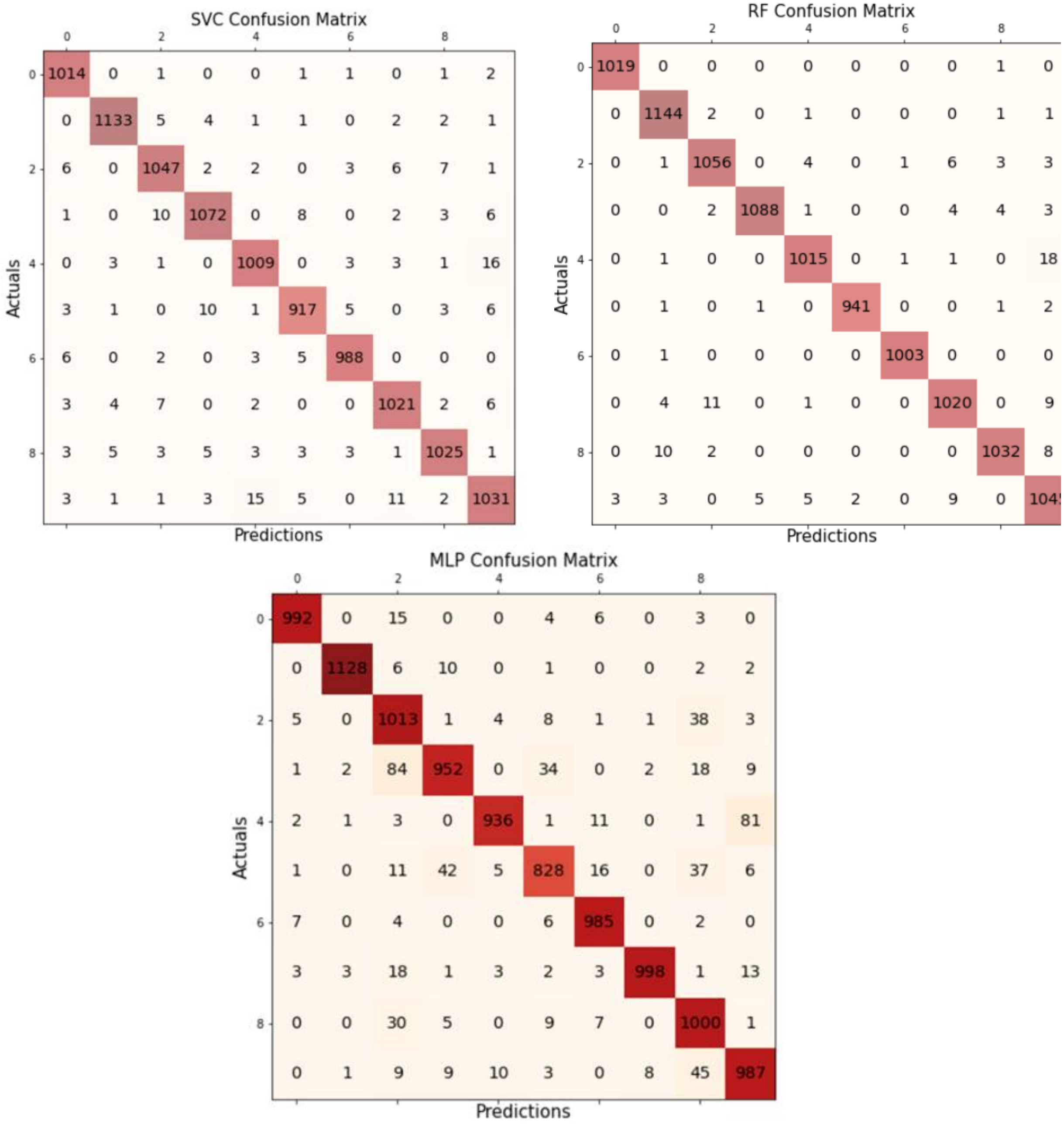
```
              precision    recall  f1-score   support

           0       1.00      0.96      0.98      1043
           1       0.98      0.99      0.98      1159
           2       0.96      0.97      0.96      1049
           3       0.94      0.97      0.95      1099
           4       0.96      0.97      0.97       990
           5       0.99      0.96      0.97       955
           6       0.98      0.98      0.98      1058
           7       1.00      0.95      0.97      1112
           8       0.92      0.99      0.95       997
           9       0.96      0.92      0.94      1038

    accuracy                           0.97     10500
   macro avg       0.97      0.97      0.97     10500
weighted avg       0.97      0.97      0.97     10500
```

Figure 16. MLP statistics

Confusion matrices for the models are provided in Figure 17.



SVC Confusion Matrix

	0		2		4		6		8	
0	1014	0	1	0	0	1	1	0	1	2
	0	1133	5	4	1	1	0	2	2	1
2	6	0	1047	2	2	0	3	6	7	1
	1	0	10	1072	0	8	0	2	3	6
4	0	3	1	0	1009	0	3	3	1	16
	3	1	0	10	1	917	5	0	3	6
6	6	0	2	0	3	5	988	0	0	0
	3	4	7	0	2	0	0	1021	2	6
8	3	5	3	5	3	3	3	1	1025	1
	3	1	1	3	15	5	0	11	2	1031

RF Confusion Matrix

	0		2		4		6		8	
0	1019	0	0	0	0	0	0	0	1	0
	0	1144	2	0	1	0	0	0	1	1
2	0	1	1056	0	4	0	1	6	3	3
	0	0	2	1088	1	0	0	4	4	3
4	0	1	0	0	1015	0	1	1	0	18
	0	1	0	1	0	941	0	0	1	2
6	0	1	0	0	0	0	1003	0	0	0
	0	4	11	0	1	0	0	1020	0	9
8	0	10	2	0	0	0	0	0	1032	8
	3	3	0	5	5	2	0	9	0	104

MLP Confusion Matrix

	0		2		4		6		8	
0	992	0	15	0	0	4	6	0	3	0
	0	1128	6	10	0	1	0	0	2	2
2	5	0	1013	1	4	8	1	1	38	3
	1	2	84	952	0	34	0	2	18	9
4	2	1	3	0	936	1	11	0	1	81
	1	0	11	42	5	828	16	0	37	6
6	7	0	4	0	0	6	985	0	2	0
	3	3	18	1	3	2	3	998	1	13
8	0	0	30	5	0	9	7	0	1000	1
	0	1	9	9	10	3	0	8	45	987

Figure 17. Confusion matrices for SVM, Random Forest and MLP

Model Limitations

The data set we used consists of two files. One file is for training the model and the second file is for testing. The testing file does not include a label column that represents the actual value of digits. Thus, we cannot use the testing dataset for evaluation and only for predictions.

The data set is mostly preprocessed, and images are converted into columns. Where each column represents a pixel of image and value indicates color of that pixel. Our models and script are based on provided dataset thus cannot process images and predict directly using images. We will need to implement this functionality in the future.

Conclusion and Future Directions

We were able to very accurately recognize digits using these models, however we will need to use a more structured data set with actual values included in order to better evaluate our model.

For future studies, we will work on improving our models and implanting functionality to process images and predict values using images. Once the testing dataset is updated and actual values are included, we will test out models against new data and re-evaluate the performance of each model.

Appendix I - Supplemental Figures for Part I

Table 6. Statistics for Decision Tree Classifier

Statistics for Decision Tree Classifier				
	Precision	Recall	F1-score	Support
Not Top 10 League	0.96	0.98	0.97	53590
Top 10 League 10	0.69	0.51	0.59	4310
Accuracy			0.95	57900
Macro Average	0.82	0.75	0.78	57900
Weighted Average	0.94	0.95	0.94	57900
Model Log Loss	0.4095			

Table 7. Statistics for Random Classifier

Statistics for Random Forest Classifier				
	Precision	Recall	F1-score	Support
Not Top 10 League	0.95	0.99	0.97	53590
Top 10 League	0.81	0.35	0.49	4310
Accuracy			0.95	57900
Macro Average	0.88	0.67	0.73	57900
Weighted Average	0.94	0.95	0.94	57900
Model Log Loss	0.1442			

Table 8. Statistics for MLP Classifier

Statistics for MLP Classifier				
	Precision	Recall	F1-score	Support
Not Top 10 League	0.95	0.98	0.97	53590
Top 10 League	0.69	0.48	0.57	4310
Accuracy			0.95	57900
Macro Average	0.83	0.73	0.77	57900
Weighted Average	0.94	0.95	0.94	57900
Model Log Loss	0.1389			

Table 9. Statistics for Gaussian NB

Statistics for Gaussian NB				
	Precision	Recall	F1-score	Support
Not Top 10 League	0.95	0.79	0.87	53590
Top 10 League	0.17	0.53	0.26	4310
Accuracy			0.77	57900
Macro Average	0.56	0.66	0.56	57900

Weighted Average	0.90	0.77	0.82	57900
Model Log Loss	1.1836			

Table 10. Statistics for Logistic Regression

Statistics for Logistic Regression				
	Precision	**Recall**	**F1-score**	**Support**
Not Top 10 League	0.94	0.99	0.96	53590
Top 10 League	0.67	0.18	0.28	4310
Accuracy			0.93	57900
Macro Average	0.80	0.58	0.62	57900
Weighted Average	0.92	0.93	0.91	57900
Model Log Loss	0.1834			

Table 11. Statistics for Voting Classifier

Statistics for Voting Classifier				
	Precision	**Recall**	**F1-score**	**Support**
Not Top 10 League	0.96	0.99	0.97	53590
Top 10 League 10	0.76	0.45	0.57	4310
Accuracy			0.95	57900
Macro Average	0.86	0.72	0.77	57900
Weighted Average	0.94	0.95	0.94	57900
Model Log Loss	0.1928			

Table 12. Statistics for Ada Boost Classifier

Statistics for Ada Boost Classifier				
	Precision	**Recall**	**F1-score**	**Support**
Not Top 10 League	0.95	0.99	0.97	53590
Top 10 League 10	0.70	0.34	0.46	4310
Accuracy			0.94	57900
Macro Average	0.83	0.67	0.71	57900
Weighted Average	0.93	0.94	0.93	57900
Model Log Loss	0.6496			

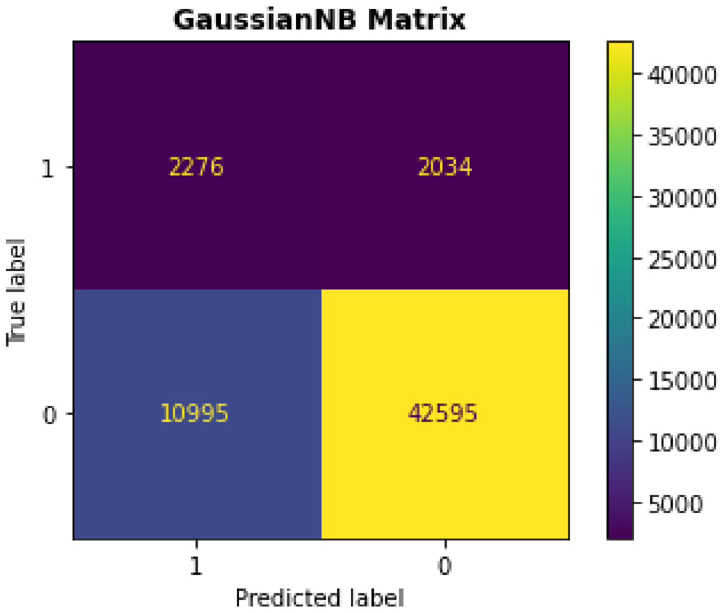


Figure 19. Gaussian NB Matrix

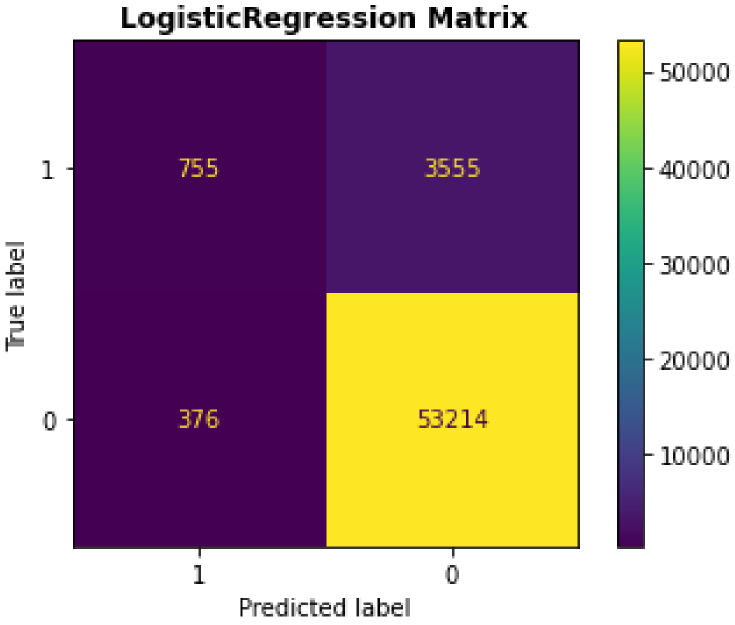


Figure 20. Logistic Regression Matrix

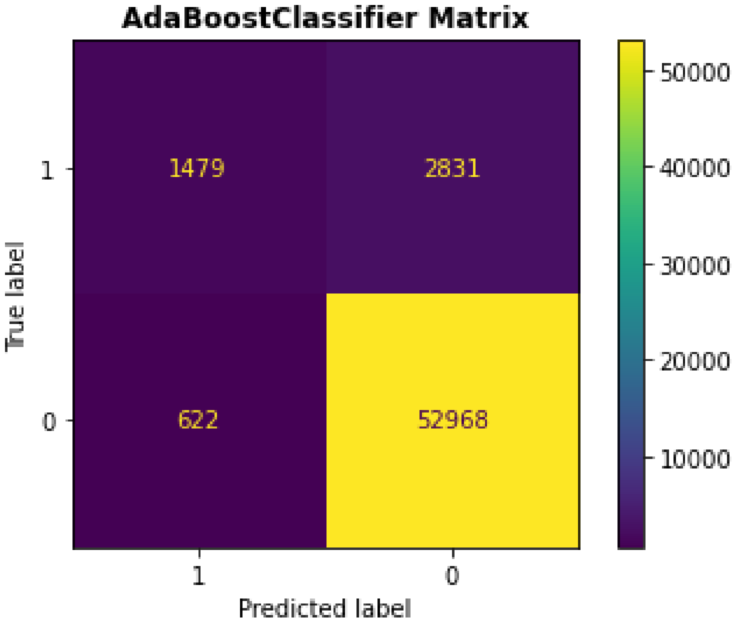


Figure 21. Ada Boost Classification Matrix