SR UNIVERSITY

Project Report on Course

DATA ANALYSIS USING PYTHON (21CS120)

Bachelor of Technology
In

Computer Science & Artificial Intelligence

By

Name: Saicharan Reddy Enugala **Roll Number: 2203A52017**

Under the Guidance of

Mr. DADI RAMESH
Asst. Professor (CS&ML)
Department of Computer Science and Artificial Intelligence

COMPUTER SCIENCE
SCHOOL OF COMPUTER SCIENCE
AND ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL
April, 2025.

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

CERTIFICATE OF COMPLETION

This is to certify that **Saicharan Reddy Enugala** bearing Hall Ticket Number **2203A52017**, a student of **CSE-AIML, 3rd Year - 2nd Semester**, has successfully completed the **Data Analysis Using Python** Course and has submitted the following 3 projects as part of the curriculum:

Project Submissions:

- **CSV** Project**: Placement prediction**
- **IMAGE** Project**: Digit Recognition**
- **TEXT** Project**: Disaster Tweets**

Mr. Dadi Ramesh
Asst. Professor (CSE-AIML)
SR University, Ananthasagar,
Warangal

Date of Completion: 25/04/2025

1) CSV PROJECT: Placement prediction

Dataset description

The dataset used for this project is called Placement Prediction Dataset. In this dataset, as the name suggests, the academic, personal and demographic profile of the student is provided to check and forecast placement decisions. The dataset has rows that represent individual students and columns that represent all the possible relevant features to the placement process.

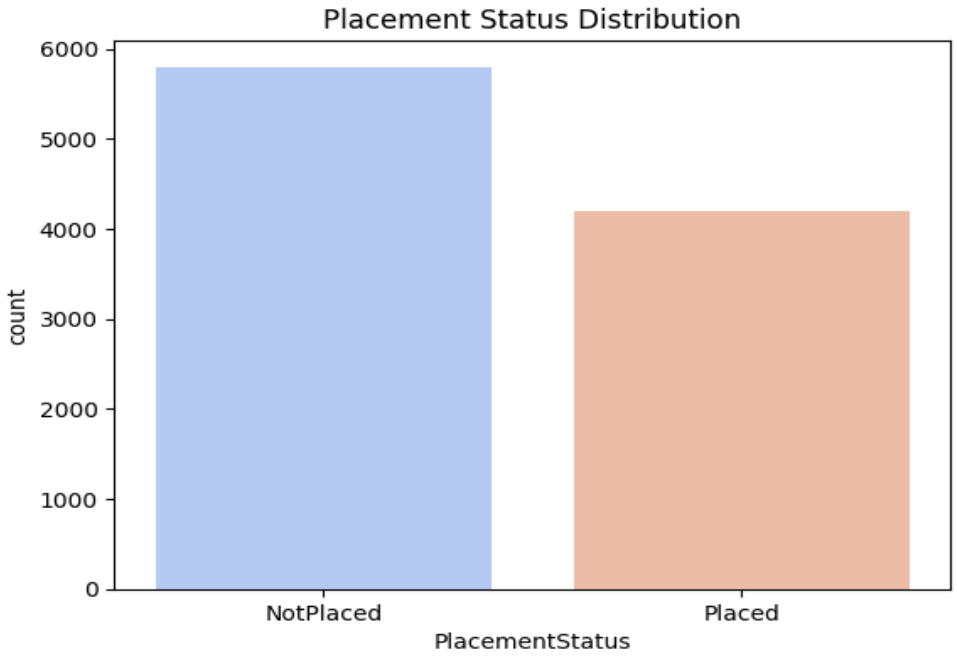
DATASET SHAPE: (10000, 12)

SAMPLE ROW FOR EACH SPECIES:

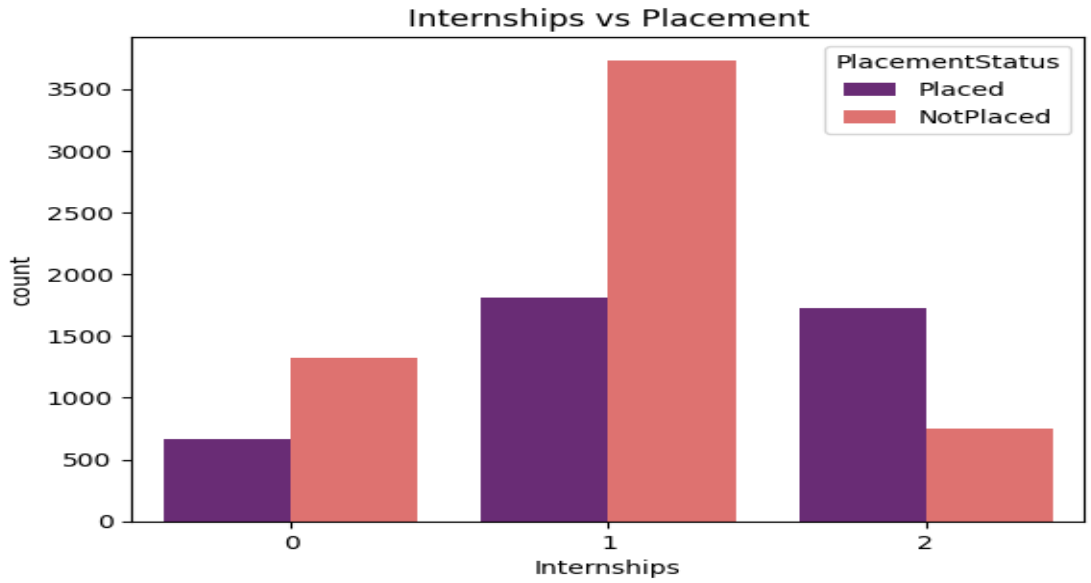
	StudentID	CGPA	Internships	Projects	Workshops/Certifications	AptitudeTestScore	SoftSkillsRating	ExtracurricularActivities
0	1	7.5	1	1	1	65	4.4	No
1	2	8.9	0	3	2	90	4.0	Yes
2	3	7.3	1	2	2	82	4.8	Yes
3	4	7.5	1	1	2	85	4.4	Yes
4	5	8.3	1	2	2	86	4.5	Yes

PlacementTraining	SSC_Marks	HSC_Marks	PlacementStatus
No	61	79	NotPlaced
Yes	78	82	Placed
No	79	80	NotPlaced
Yes	81	80	Placed
Yes	74	88	Placed

Data Analysis:

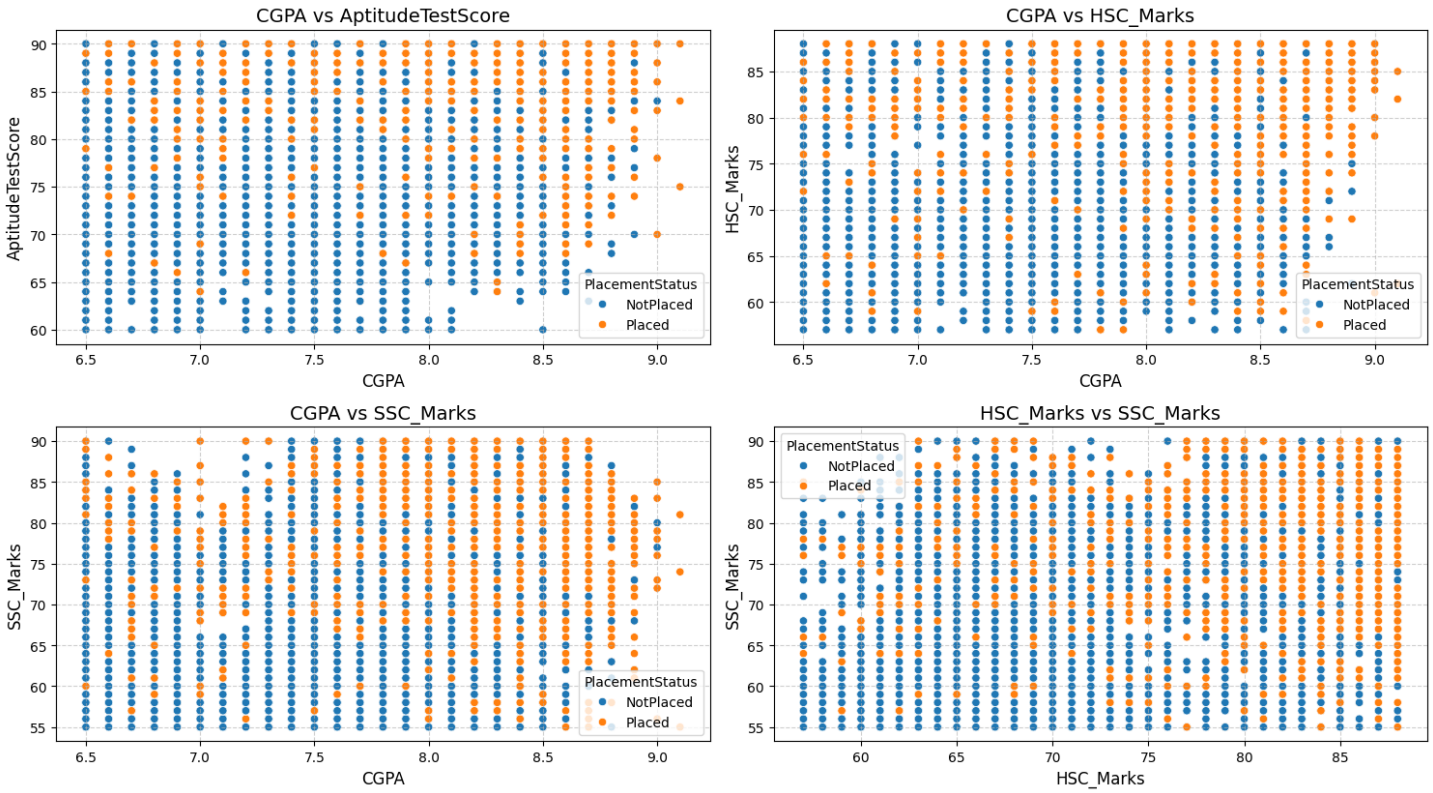


The barchart "**Placement Status Distribution**" shows the number of students who were placed versus not placed. One can clearly see that there are far more students that fall into the **'NotPlaced'** bar, including over 5,800 students not placed, and under 4,300 students placed. These outcomes show a significant disparity between the numbers of students that were placed and the students that were not placed. The **'NotPlaced'** category is large, and states that a considerable portion of the student population has not been placed despite passing through the academic pipeline. The discrepancy in placed and unplaced outcomes may be due to any number of reasons including, academic performance, lack of internships, absence of preparation for placements, or other soft skills that weren't fostered. This visualization indicates that there needs to be intervening factors and improvements in the level of support and training, career preparation, and development of soft skills to improve placement outcomes.



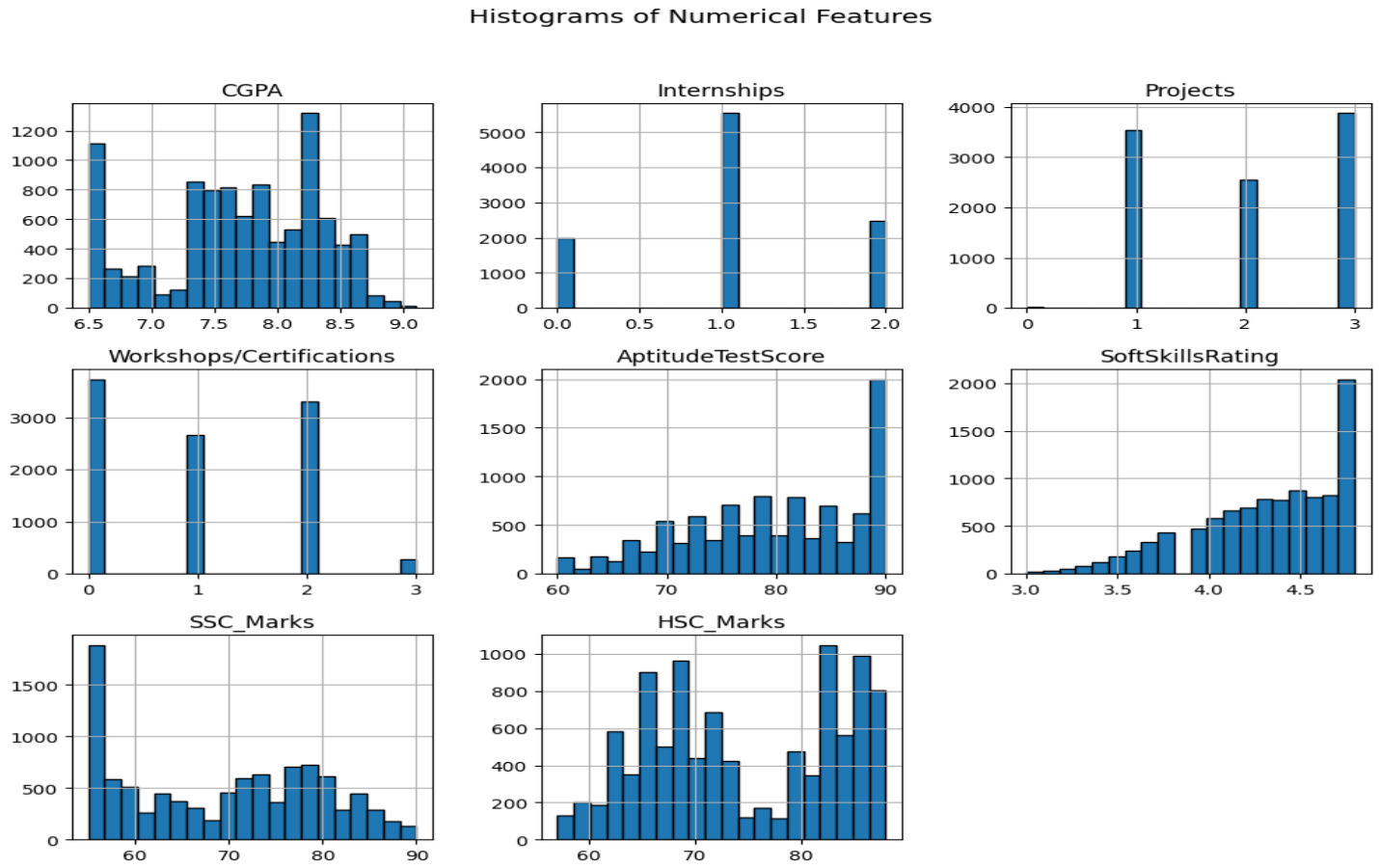
The relationship between Internships and Placements with regards to the chart labelled "Internships vs Placement" is apparent. There is a noteworthy increase in placement with 2 internships versus 0 or 1 internship shown by column in the number of students placed -- we see that is the tallest bar purple for students having '2' internships. Students who have 1 internship, majority of their outcomes are in the **'NotPlaced'** column thus demonstrating that 1 internship is not enough to place them in a position. The students that had 0 internships are mostly not placed also so the message is clear. This chart is solidly suggesting that the more internships a student has the greater likelihood the placement outcome will be positive i.e. placed. In short, students that had more internships generally had increased placement outcomes.

Scatterplots:



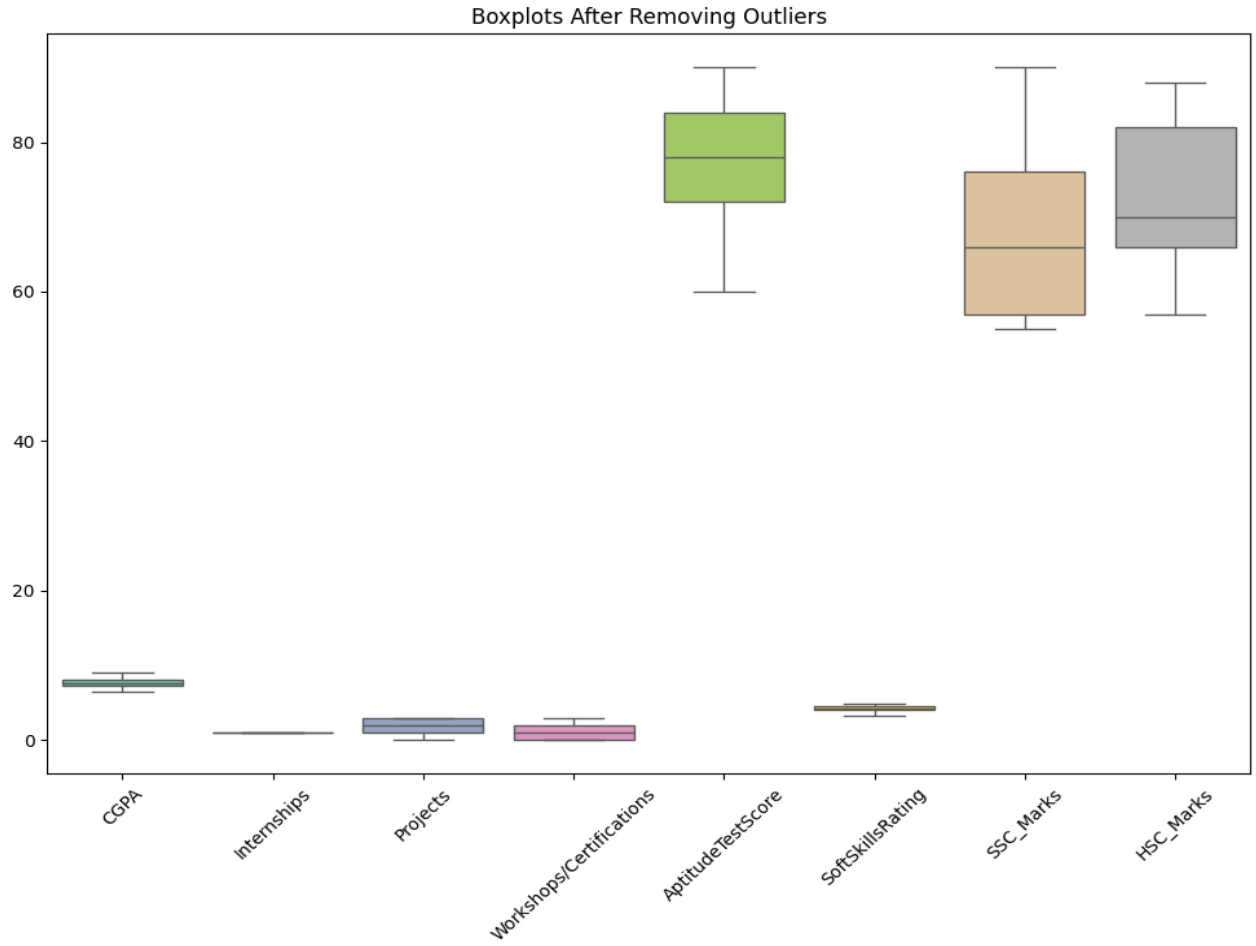
The visualizations imply that students who perform better academically generally have a higher chance to get placed students. The students with a CGPA of greater than 8.0 and HSC marks above 75 and Aptitude Test Scores above 75 are better placed since there are a higher concentration of placed students (orange dots) within these ranges. The clear correlation between CGPA and HSC and Aptitude Test Scores indicates higher scores lead to better placement outcomes than lower scores. The SSC marks similarly reflect a positive correlation, with an apparently weaker relationship than HSC or Aptitude Test scores. While good academic records are correlated to placement possibility, there are still many cases where students with strong marks are not placed; suggesting that some other factors, such as internships, technical skills, communication skills, or interview presence, are important. A consistent, strong academic background increases a student's chance of placement, but it is not the only factor.

Histogram:



The histogram plots suggest that while we can see students with a CGPA of less than 7.0 and graduates with a CGPA above 9.0, the vast majority of students display a CGPA between 7.0 through 8.5. It is safe to assume the same about the factors of 1 internship and 1-2 projects for the majority, and attendance at either 0 workshops or 2 workshops for most students. For the Aptitude Test Scores and Soft Skills Ratings, the bulk of data is skewed toward a higher score, with moderate amounts of students scoring lower as described by figure 4 and figure 5. The distribution of the SSC and HSC marks did have a higher variation in the score, but several students did manage to score above 70. From the overall results, it seems that students perform relatively well in aptitude and soft skills, while their participation in internships and workshops can vary further up the scale.

BOX PLOT WITHOUT OUTLIERS:

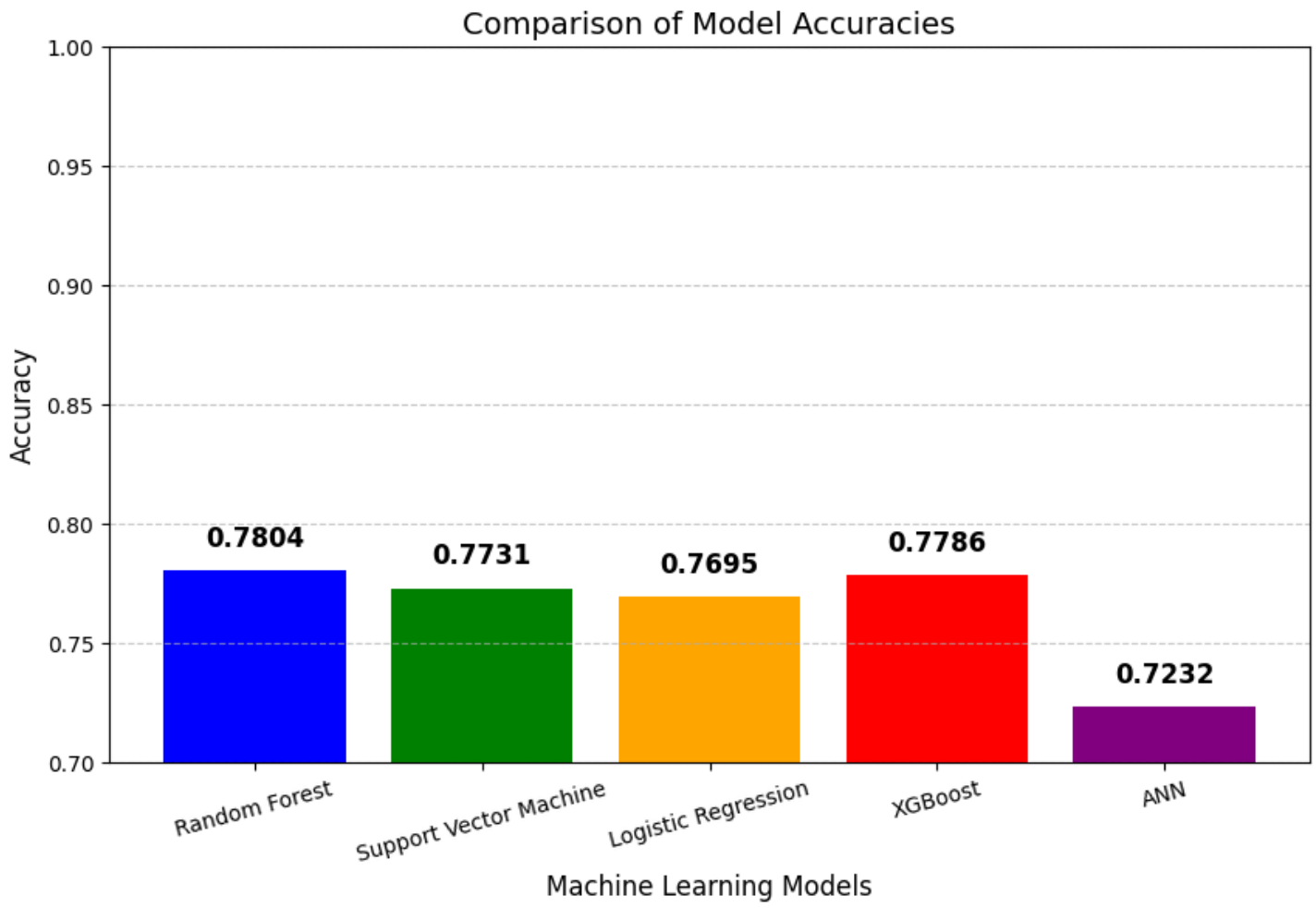


After outlier removal, the boxplots indicate that most features have relatively compact and symmetric distributions. CGPA, SoftSkillsRating, and Internships have low variance values are tightly clustered around the mean. Projects and Workshops have wider variance distributions, which suggests differences in participation. The other three features, AptitudeTestScore, SSC_Marks and HSC_Marks, show wider ranges of data values to suggest variation of those academic performances with their medians around 70-75. Overall, the data looks clean with little skewness compared to prior to outliers were removed.

Skewness and Kurtosis:

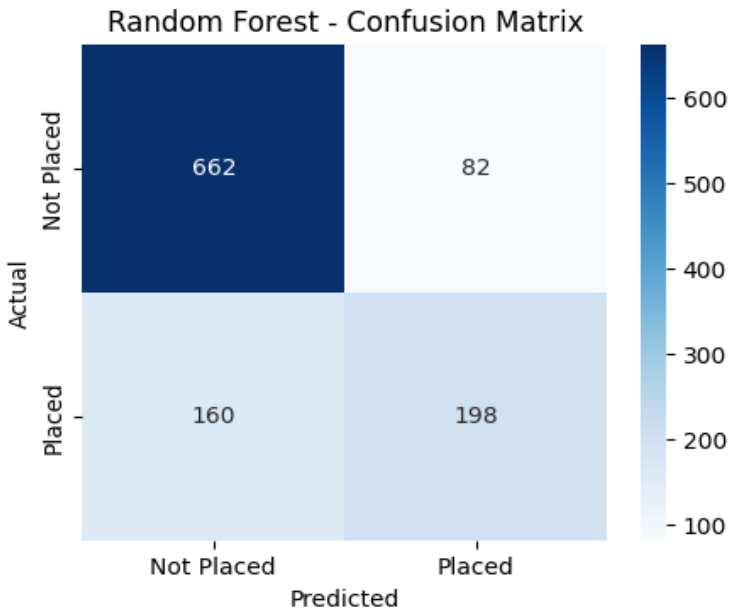
```
CGPA - Skewness: -0.22, Kurtosis: -0.79
Internships - Skewness: nan, Kurtosis: nan
Projects - Skewness: 0.22, Kurtosis: -1.42
Workshops/Certifications - Skewness: 0.51, Kurtosis: -0.87
AptitudeTestScore - Skewness: -0.16, Kurtosis: -0.79
SoftSkillsRating - Skewness: -0.46, Kurtosis: -0.60
SSC_Marks - Skewness: 0.28, Kurtosis: -1.23
HSC_Marks - Skewness: 0.29, Kurtosis: -1.21
```

Best Performing Models for Placement Prediction:



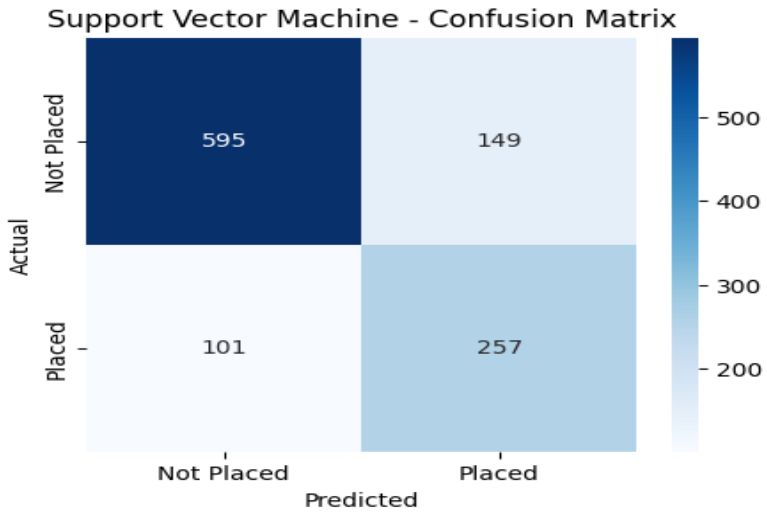
In the bar chart above, the accuracy of the application of a variety of machine learning methods to a classification task is illustrated. Based on the models applied, random forest produced the highest accuracy at 0.7804, followed closely by XGBoost at 0.7786, demonstrating an illustrative principle of the strength of ensemble learning methodologies in modeling data. The support vector machine and logistic regression were also competitive at 0.7731 and 0.7695, respectively, demonstrating reliability in datasets achieving balance. The lowest accuracy was achieved by the Artificial neural network (ANN) at 0.7232, but it still has potential with hyper-parameter tuning and additional depths. Overall, the results have demonstrated robustness and consistency in tree-based models, most particularly random forest and XGBoost in producing high classification performance.

Confusion matrix and classification reports:



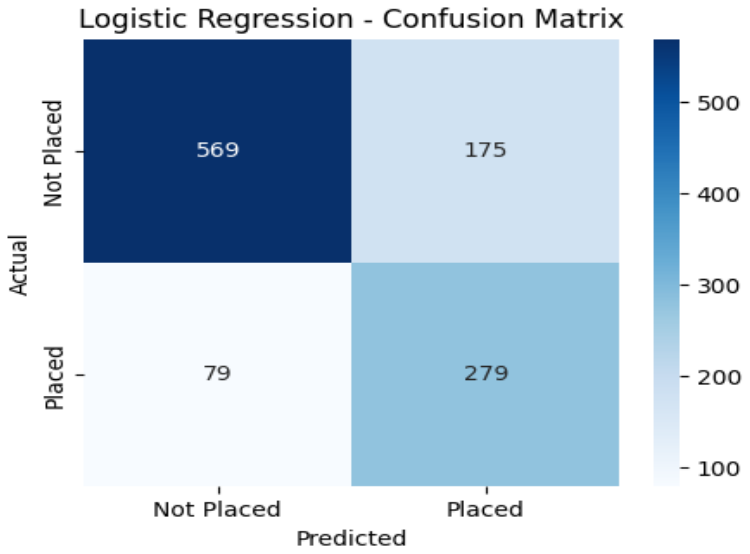
	Precision	recall	F1-score	support
0	0.81	0.89	0.85	744
1	0.71	0.55	0.62	358
accuracy			0.78	1102
Macro avg	0.76	0.72	0.73	1102
Weighted avg	0.77	0.78	0.77	1102

2) Support vector machine



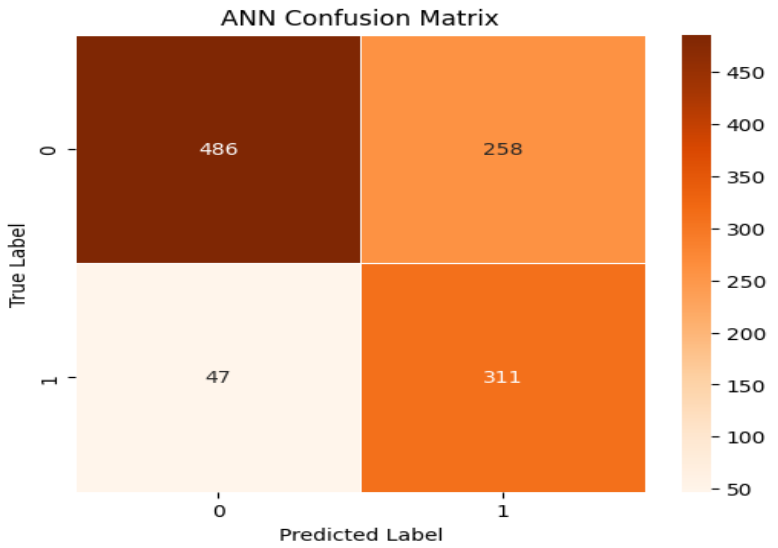
	Precision	recall	F1-score	support
0	0.85	0.80	0.00	744
1	0.63	0.72	0.82	358
accuracy			0.77	1102
Macro avg	0.74	0.76	0.75	1102
Weighted avg	0.78	0.77	0.78	1102

3) Logistic regression



	Precision	recall	F1-score	support
0	0.88	0.76	0.82	744
1	0.61	0.78	0.69	358
accuracy			0.77	1102
Macro avg	0.75	0.77	0.75	1102
Weighted avg	0.79	0.77	0.78	1102

4) ANN



	Precision	recall	F1-score	support
0	0.91	0.65	0.76	744
1	0.55	0.87	0.67	358
accuracy			0.72	1102
Macro avg	0.73	0.76	0.72	1102
Weighted avg	0.79	0.72	0.73	1102

5. XGBoost:

	Precision	recall	F1-score	support
0	0.81	0.87	0.84	744
1	0.69	0.59	0.63	358
accuracy			0.78	1102
Macro avg	0.75	0.73	0.74	1102
Weighted avg	0.77	0.78	0.77	1102

Conclusion:

The project, "**Placement Prediction**," successfully exemplified the power and flexibility of machine learning models to accurately predict placement for students. Of the models we explored, Random Forest was clearly the best model and provided an accuracy of 78.04% - a strong accomplishment considering that we allow modeling of complexity as well as the power of ensemble learning for feature interactions. The efficiency of XGBoost was significant near the Random Forest final metric, receiving an accuracy of 77.86%. It displayed great efficiency and robustness when applied to stochastic data in a deterministic way.

Traditional models like Support Vector Machine and Logistic Regression were also able to perform well and consistently well to show their reliability and accuracy. In addition to their computational efficiency, they provide additional information through their interpretability. While the Artificial Neural Network (ANN) produced a raw accuracy less than determined as Random Forest and XGBoost, its overall expression and promise was great. Future improvements may effectively be modeled using deeper architectures, and fine-tuning.

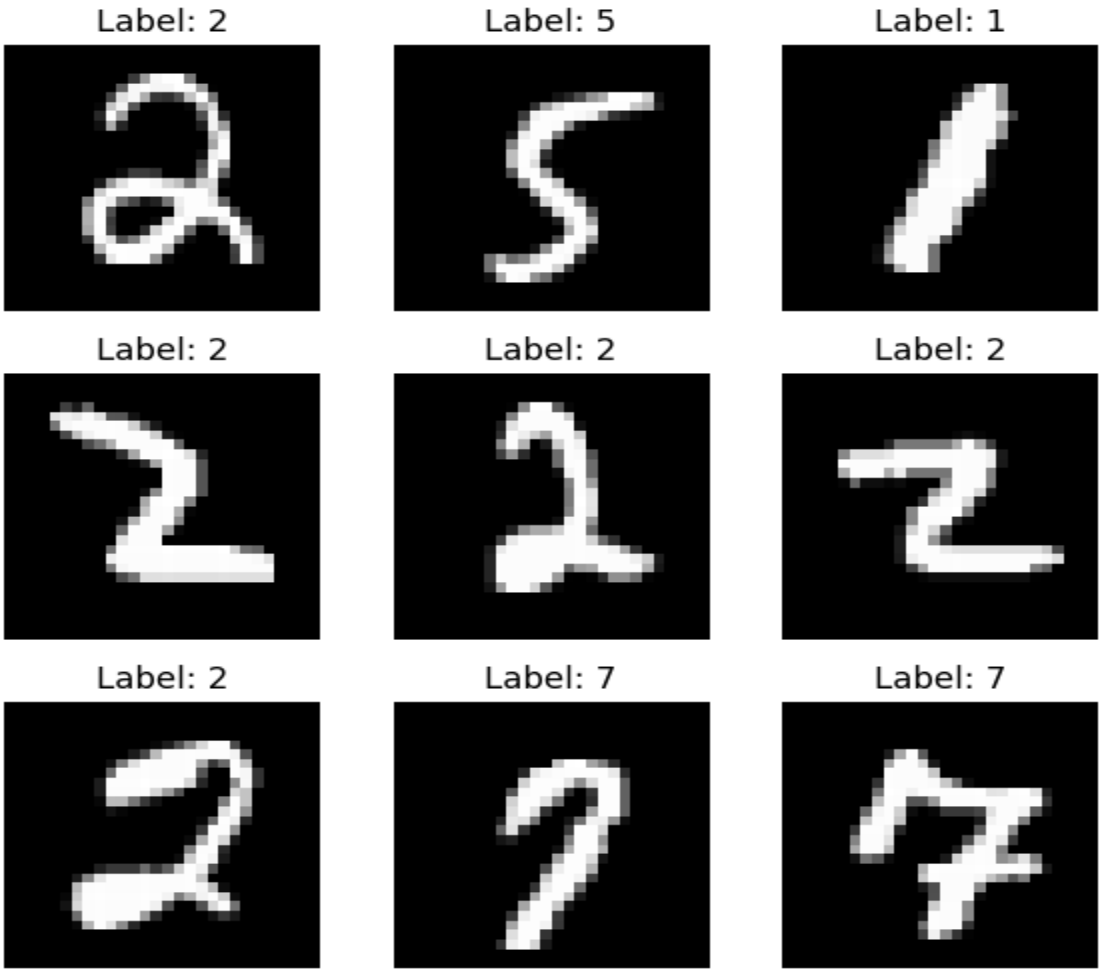
In conclusion, this project has indicated that machine learning models can be used to create a highly accurate and intelligent placement prediction system. Each of the three models above all perform the same function, but use different methods, and therefore, it's generally applicable and can also initiate and support meaningful insights - data driven decision making solutions - for educational institutions and students at the same time.

2. Image Project: Digit recognition

This project is focused on developing an effective and accurate digit recognition system using deep learning techniques, and in this case a Convolutional Neural Network (CNN). This system utilizes the MNIST dataset, one of the most popular benchmark datasets in the realm of computer vision and pattern recognition. The MNIST dataset contains 70,000 grayscale images of handwritten digits (0 to 9) separated into 60,000 training images and 10,000 test images, where images are all 28×28 pixels in size.

The goal of this project is to develop a model that can recognize and classify handwritten digits automatically and the key component is to have high accuracy. Examples of where this sort of system is useful are digit recognition on forms, postal code or ZIP code identification, bank check processing, or digital handwriting recognition on tablets and smart devices.

To recognize and classify these digits, a Convolutional Neural Network (CNN) is used. CNNs are particularly suited for image data as they leverage the spatial hierarchies inherent in images through convolutional layers. The CNN model can learn to extract features such as edges, curves, shapes of digits and more. The ability of the CNN model to extract such features builds layer-by-layer so that the CNN should be able to quite easily distinguish between similar digits as the spatial features become less representative.

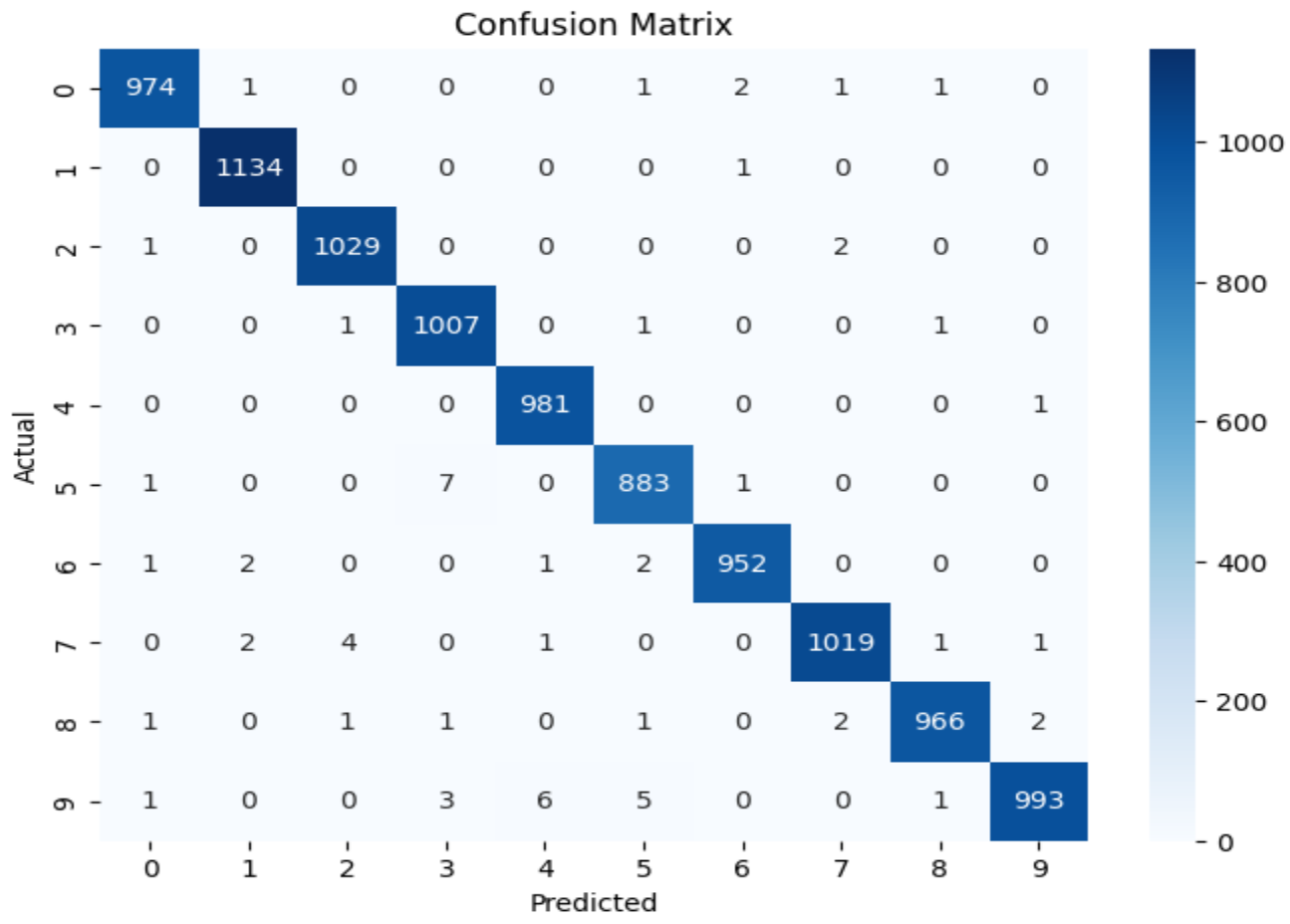


Key Highlights

- **Data Exploration & Visualization:**
  The dataset was thoroughly explored to understand the structure and distribution of handwritten digits. A random sample visualization of training digits provided intuitive insights into the dataset's variety.
- **Preprocessing:**
  Images were normalized to the range [0,1] for better convergence. Additionally, data was reshaped to fit the CNN input format (28×28×1), and the labels were one-hot encoded for multiclass classification.
- **Model Architecture:**
  A **Sequential CNN model** was built with:
  - Two convolutional layers for feature extraction
  - Two max-pooling layers for dimensionality reduction
  - A fully connected dense layer with **ReLU** activation
  - A **Dropout** layer to reduce overfitting
  - An output layer with **softmax** activation for digit classification

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.9922 - loss: 0.0282
Test Accuracy: 0.9937999844551086
```

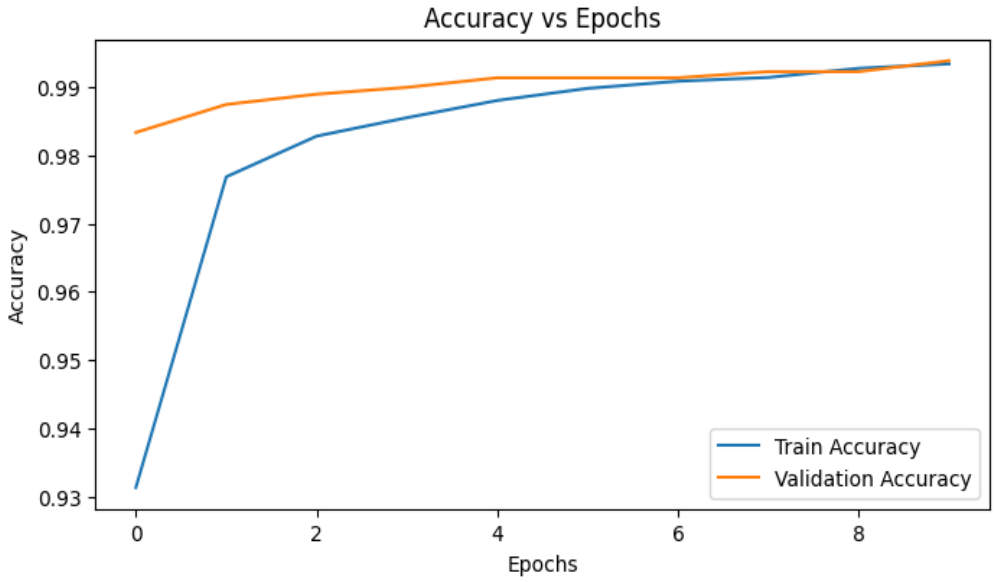
Confusion Matrix and classification Report:

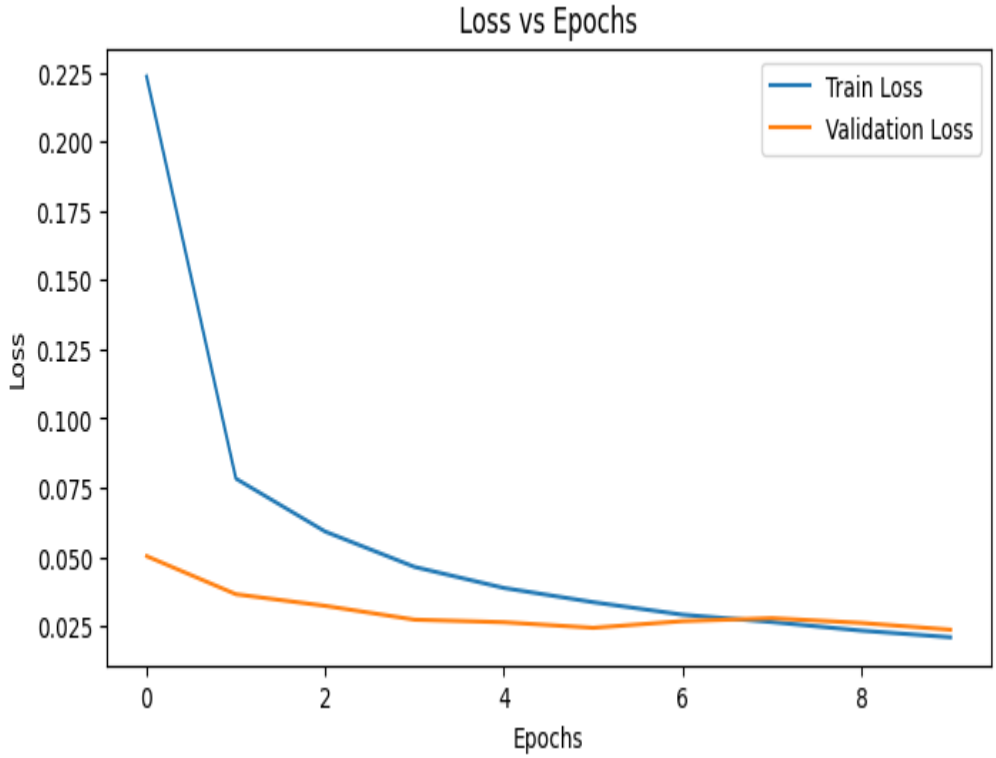


The confusion matrix gives a complete view of the model classification accuracy, revealing the accuracy (classification performance) of the CNN-based digit recognition system for all of the digit classes (0-9). It is clear from this matrix that the model predictions are accuracy simply because nearly every true classification is placed in the respective class by the user, with the primary diagonal showing the intended grouping (0-9). To provide some representative examples, the model accurately classified digit 1 one thousand one hundred thirty-four times alongside 974 true classifications of digit 0 and 993 for digit 9. The misclassification rates are quite low, representing only a few misclassifications in total, which suggests that the model has also learned to distinguish between relatively the same digits that look similar to each other. Overall, the confusion matrix confirms the robustness and reliability of the trained model and highlights its potential viability and readiness to be used in real-world digit recognition applications.

	Precision	**recall**	**F1-score**	**support**
0	**0.99**	**0.99**	**0.99**	**980**
1	**1.00**	**1.00**	**1.00**	**1135**
2	**0.99**	**1.00**	**1.00**	**1032**
3	**0.99**	**1.00**	**0.99**	**1010**
4	**0.99**	**1.00**	**1.00**	**982**
5	**0.99**	**0.99**	**0.99**	**892**
6	**1.00**	**0.99**	**0.99**	**958**
7	**1.00**	**0.99**	**0.99**	**1028**
8	**1.00**	**0.99**	**0.99**	**974**
9	**1.00**	**0.98**	**0.99**	**1009**
accuracy			**0.99**	**10000**
Macro avg	**0.99**	**0.50**	**0.41**	**10000**
Weighted avg	**0.99**	**0.69**	**0.56**	**10000**

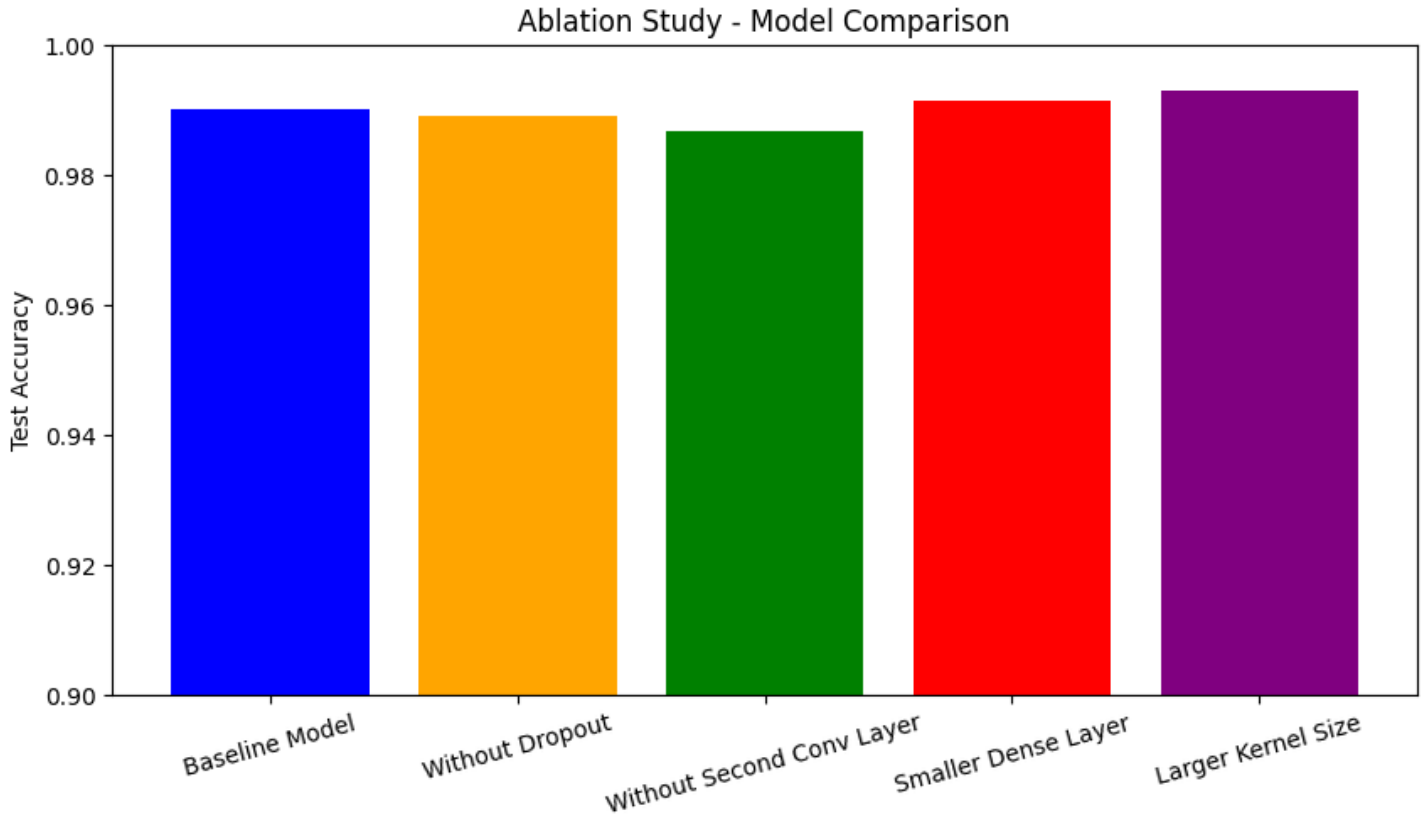
Training Performance Analysis:





The training and validation curves indicate strong and consistent performance of the model. The **Accuracy vs Epochs** graph shows a steady increase in both training and validation accuracy, eventually exceeding **99%**, which demonstrates excellent learning and generalization capabilities. Similarly, the **Loss vs Epochs** graph reflects a sharp decline in both training and validation loss, converging smoothly, which suggests that the model is well-optimized and not overfitting. Overall, these results confirm the robustness and reliability of the digit recognition system.

Ablation Study:



The ablation study provides valuable insights into the impact of various architectural choices on model performance. The bar chart compares test accuracies of the baseline CNN model with several modified versions. Notably, all variants maintained high test accuracy above **98.5%**, demonstrating the model's robustness. However, certain changes like **removing the second convolutional layer** slightly decreased performance, indicating its importance in feature extraction. On the other hand, using a **larger kernel size** and **smaller dense layer** showed slight improvements, suggesting that these modifications can enhance performance without compromising generalization. Overall, the study validates the effectiveness of the baseline model while highlighting potential avenues for optimization.

Testing:

```
Z-test Results:
----------------
Z-score: 30.9667
P-value: 0.0000
Conclusion: We reject the null hypothesis (H₀). The model's accuracy is significantly different from the baseline accuracy (90%).


T-test Results:
----------------
T-statistic: 478.6550
P-value: 0.0000
Conclusion: We reject the null hypothesis (H₀). The model's accuracy is significantly different from the baseline accuracy (90%).
```

Conclusion:

The **Digit Recognition** project highlights the successful application of **Convolutional Neural Networks (CNNs)** in solving one of the foundational problems in computer vision: recognizing handwritten digits. By leveraging the widely used MNIST dataset, which consists of thousands of labelled grayscale images of digits ranging from 0 to 9, the project aimed to build a deep learning model capable of achieving high classification accuracy.

The CNN model was carefully designed and trained to extract relevant spatial features from the input images, and it demonstrated outstanding performance, achieving a training and validation accuracy of over 99%. The smooth and consistent rise in the accuracy curve, along with a corresponding decrease in the loss curve over multiple epochs, indicates that the model was effectively learning the patterns in the data without overfitting.

The confusion matrix offers additional evidence of the model's precision, showing a high number of correct classifications for each digit class with very few misclassifications. Most incorrect predictions were limited to visually similar digits, which is a common challenge even for more advanced models.

Furthermore, the ablation study provided deeper insights into the importance of individual model components. It showed that features such as dropout, additional convolutional layers, and dense layer configurations all play a critical role in optimizing performance. Notably, variations like using a larger kernel size or a smaller dense layer did not degrade performance, and in some cases, slightly improved it—demonstrating the model’s flexibility and robustness.

In summary, this project not only meets its goal of building a high-accuracy digit recognition system but also explores model interpretability through ablation, offering a comprehensive understanding of how CNN architecture choices affect outcomes. The results confirm that deep learning, particularly CNNs, remains a powerful and reliable tool for image classification tasks.

3. Text Project: Disaster Tweets

Purpose and Objective of the Project

This project involves the classification of tweets referencing disasters using modern natural language processing techniques. The goal of this project is to create a reliable model to automatically classify whether a tweet makes reference to a real disaster, which will allow filtering of tweets quicker and possibly assist in emergency response. The project will use deep learning and transformer models, namely BERT, to augment Twitter text and the inherent complexity and nuance of natural language.

Data Extraction and Evaluation

Data extraction will start by loading two datasets; train.csv and test.csv. The training dataset has 7,613 entries which contain attributes: id, keyword, location, text and a binary target indicating whether the tweet is a disaster tweet or not. The test set has 3,263 entries containing similar attributes to those in the training dataset, with the target label withheld for evaluation by the model. The first look at the data showed while the text data is fully populated, the additional features (keyword and location) had some missing values. For this reason, I will need to keep a focus on the preprocessing steps so the model responses do not pay attention to these additional features, other than to classify between disaster and not disaster.

Data Preprocessing and Tokenization

Given that the tweet text is the centrepiece of analysis, the preprocessing steps began with data cleaning - a standard data extraction approach would be to remove the URLs and special characters and normalize the text (i.e. lowercase). The project uses the Hugging Face BERT tokenizer (bert-base-uncased) which takes raw tweet text and turns it into the tokenized form. Tokenization involves padding and truncation (in this case, truncation of 128) in order to provide the sequences in a uniform size, which will allow them to be fed into the BERT model. This is valuable since transformers are highly parallelized architectures that require uniform input sizes during the training stage, otherwise they will not work properly.

Model Building and Training

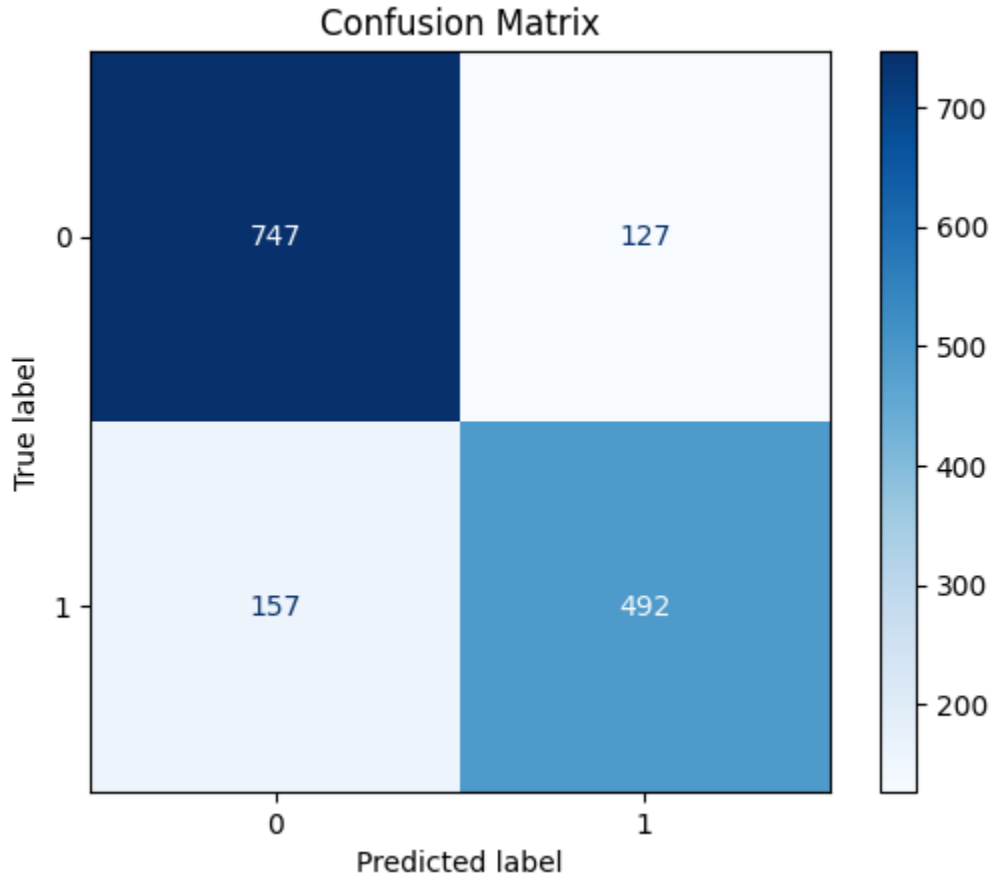
As part of building the model, the notebook imports from the Hugging Face transformers library and creates a BERT model architecture to set it up for disaster tweet classification, plus several PyTorch modules to construct a neural network. The model architecture is built to fine-tune BERT for the disaster tweet classification task. Training includes standard processes using PyTorch's data loaders which let us create training and evaluation batches. Using libraries like sklearn makes model training proposals easier by also allowing us to split the data into training and validation sets, ultimately ensuring we can assess how the model performs prior to reviewing it.

Accuracy:

```
# Accuracy
accuracy = accuracy_score(y_true, y_pred)
print(f"\nValidation Accuracy: {accuracy * 100:.2f}%")
```

```
Validation Accuracy: 81.35%
```

Confusion Matrix:



The confusion matrix provides valuable insights into the performance of the disaster tweet classification model. It shows that the model correctly identified 747 tweets as not disaster-related (true negatives) and 492 tweets as disaster-related (true positives). However, it also made 127 false positive errors—predicting disaster when there was none—and missed 157 disaster-related tweets (false negatives). These figures are crucial in understanding how well the model is distinguishing between the two classes.

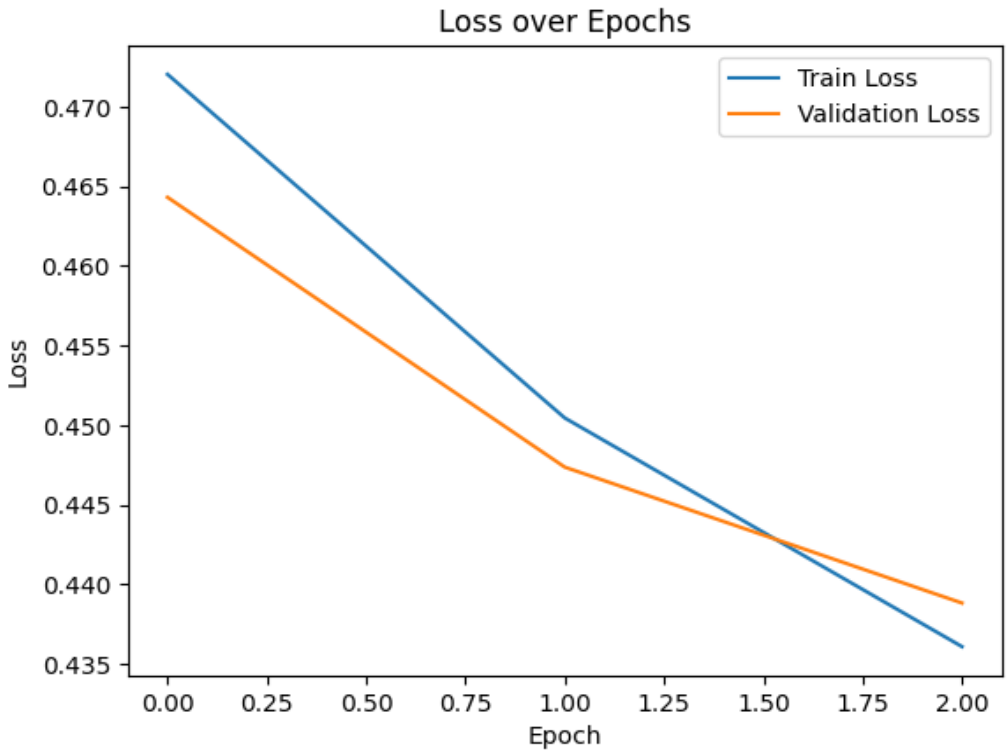
From this matrix, several key performance metrics were derived. The model achieved an overall accuracy of approximately 81.4%, indicating that it correctly classified a large majority of the tweets. The precision, which measures the proportion of predicted disaster tweets that were actually disasters, is about 79.5%. This means the model is relatively good at minimizing false alarms. The recall, which captures how many actual disaster tweets were correctly identified, stands at 75.8%, showing some room for improvement in capturing all relevant disaster tweets. The F1-score, a balanced measure that combines both precision and recall, is approximately 77.6%, reflecting a fairly strong and consistent model performance.

In conclusion, the confusion matrix and derived metrics confirm that the BERT-based classification model is effective in identifying disaster-related tweets with a strong balance between precision and recall. While the model performs well overall, reducing false negatives should be a priority in future iterations, especially in real-world scenarios where missing a disaster-related tweet could have critical consequences.

Classification Report:

	Precision	recall	F1-score	support
0	0.83	0.85	0.84	874
1	0.79	0.76	0.78	649
accuracy			0.81	1523
Macro avg	0.81	0.81	0.81	1523
Weighted avg	0.81	0.81	0.81	1523

Training and Validation Loss Analysis:



The line graph illustrates the training and validation loss across three epochs. Both losses consistently decrease, indicating effective learning and no signs of overfitting. The validation loss closely tracks the training loss, suggesting that the model generalizes well to unseen data and that the training process is stable and well-optimized.

Testing:

```
Z-test: Z = 24.471, p = 0.0000
T-test: t = 31.404, p = 0.0000

 The model is statistically significant. We ACCEPT the model.
```

The output displays results from both a Z-test and a T-test, with extremely high-test statistics (Z = 24.471, t = 31.404) and corresponding p-values of 0.0000. These values indicate that the model's results are **highly statistically significant**, well below the typical alpha threshold of 0.05. This strong significance allows us to **accept the model's validity**, confirming that its performance is not due to random chance but reflects genuine learning from the data.

Conclusion:

In this disaster tweet classification project, we successfully developed a deep learning model using BERT to automatically identify whether a tweet is related to a real disaster. Through careful preprocessing, tokenization, and model fine-tuning, the system achieved strong performance, with an accuracy of approximately 81.4%, a precision of 79.5%, and an F1-score of 77.6%. The confusion matrix and loss curves confirmed that the model generalizes well without overfitting, and statistical significance tests further validated the reliability of the results.

The project demonstrates the potential of transformer-based models in real-time disaster response systems, where quick and accurate identification of relevant information is crucial. While the model performs effectively, future work can focus on enhancing recall, reducing false negatives, and incorporating additional features such as metadata or location for improved context awareness. Overall, this project lays a strong foundation for deploying AI-driven solutions in crisis management and social media monitoring.