BUDT737 Final Project Spring 2023
Group 13: Data Squad

Madison Sanchez
Charishma Jaladi
Anirudh Murali
Shrushti Shah
Arvind Kanhirampara Ravi

Introduction

Predicting target labels based on the words used in textual data is a fundamental problem in Natural Language Processing (NLP). The ability to automatically classify text into binary categories (0 or 1) has numerous practical applications, such as sentiment analysis, spam detection, and identifying fake news. By leveraging machine learning algorithms and NLP techniques, we aim to develop a predictive model that can accurately determine the target label based on the words present in the text.

In this project, our focus is to create a robust and efficient system for predicting the target label using textual data. We will leverage a labeled dataset, where each instance is associated with a binary target label. The dataset contains a diverse range of text samples, such as customer reviews, social media posts, or product descriptions. Our objective is to build a model that can generalize well to unseen data and make accurate predictions, thus providing valuable insights and aiding decision-making processes.

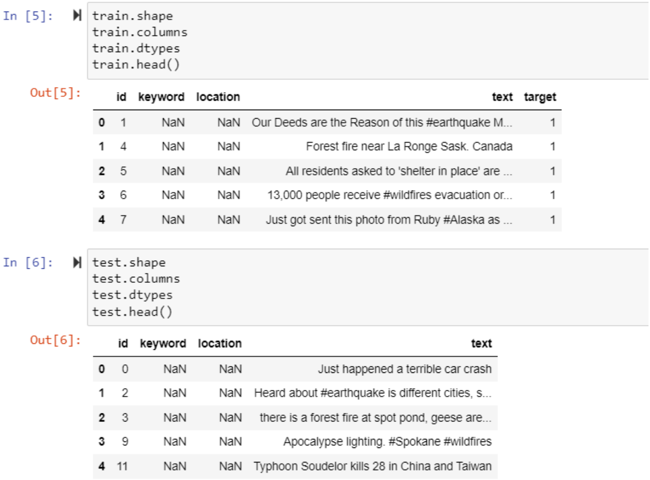
To achieve this, we will employ a combination of feature engineering techniques and machine learning algorithms. Feature engineering involves extracting meaningful information from the text, such as word frequencies, n-grams, or syntactic patterns, to represent the data in a format suitable for machine learning algorithms. Additionally, we will explore various supervised learning techniques, such as logistic regression, support vector machines, or neural networks, to train and evaluate our predictive model.

Furthermore, we will pay attention to potential challenges in this task, including handling noisy or unstructured text, addressing class imbalance issues, and mitigating the impact of bias in the data. By addressing these challenges and fine-tuning our model, we aim to achieve a high level of accuracy and robustness in predicting the target labels based on the words used in the given textual data.

Ultimately, the successful completion of this project will contribute to the advancement of NLP techniques for text classification and provide practical insights into the potential applications of such models in real-world scenarios.

Code Example

Snippet of Initial Exploratory Data Analysis:



```
In [5]:  train.shape
          train.columns
          train.dtypes
          train.head()
```

Out[5]:

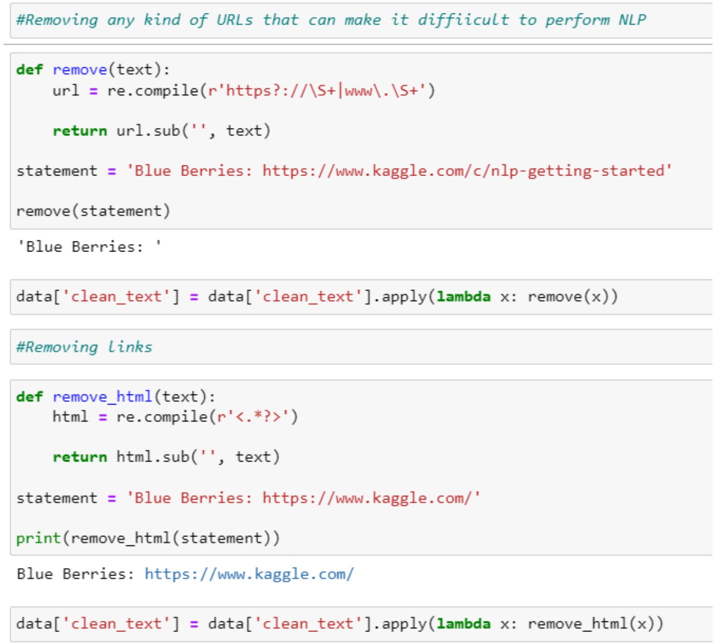
	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1
4	7	NaN	NaN	Just got sent this photo from Ruby #Alaska as ...	1

```
In [6]:  test.shape
          test.columns
          test.dtypes
          test.head()
```

Out[6]:

	id	keyword	location	text
0	0	NaN	NaN	Just happened a terrible car crash
1	2	NaN	NaN	Heard about #earthquake is different cities, s...
2	3	NaN	NaN	there is a forest fire at spot pond, geese are...
3	9	NaN	NaN	Apocalypse lighting. #Spokane #wildfires
4	11	NaN	NaN	Typhoon Soudelor kills 28 in China and Taiwan

Snippet of Data Cleaning:



```
#Removing any kind of URLs that can make it diffiicult to perform NLP
```

```
def remove(text):
    url = re.compile(r'https?://\S+|www\.\S+')

    return url.sub('', text)

statement = 'Blue Berries: https://www.kaggle.com/c/nlp-getting-started'

remove(statement)
```

```
'Blue Berries: '
```

```
data['clean_text'] = data['clean_text'].apply(lambda x: remove(x))
```

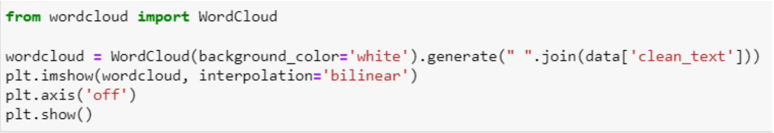
```
#Removing Links
```

```
def remove_html(text):
    html = re.compile(r'<.*?>')

    return html.sub('', text)

statement = 'Blue Berries: https://www.kaggle.com/'

print(remove_html(statement))
```

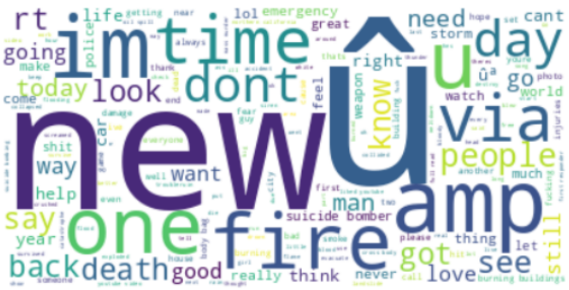
```
Blue Berries: https://www.kaggle.com/
```

```
data['clean_text'] = data['clean_text'].apply(lambda x: remove_html(x))
```

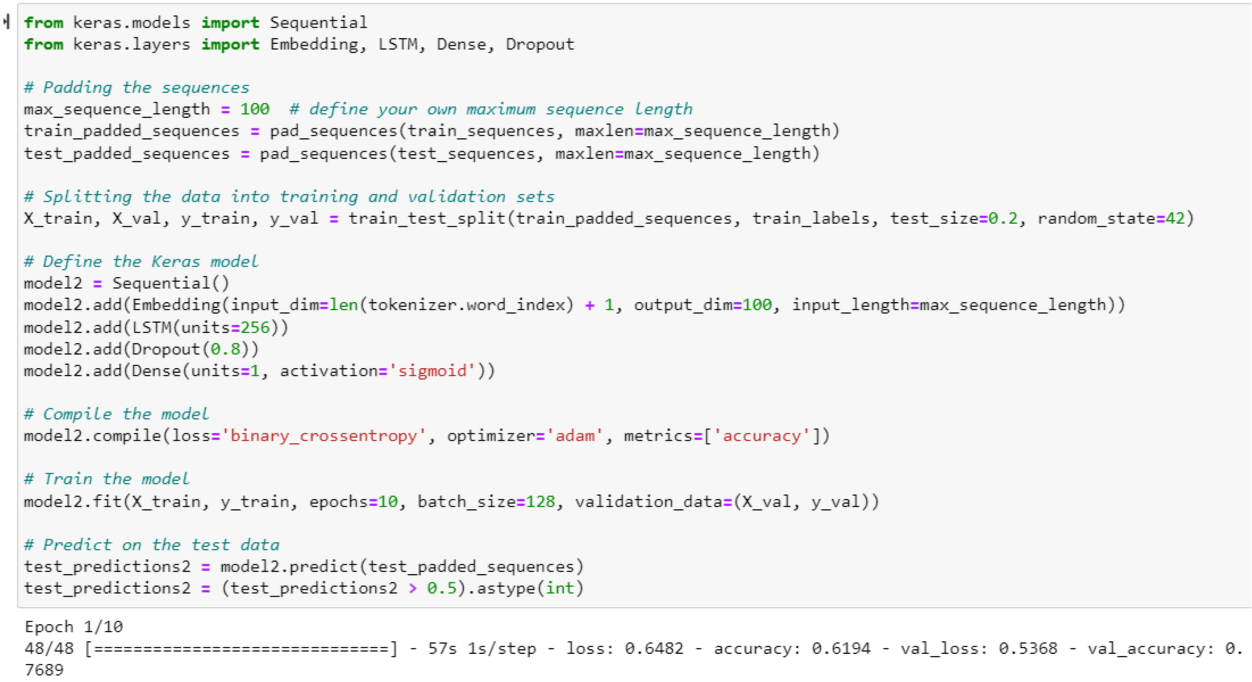
Snippet of Visualizing the words that occur the most:



```
from wordcloud import WordCloud

wordcloud = WordCloud(background_color='white').generate(" ".join(data['clean_text']))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



Snippet of Modeling:



```
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense, Dropout

# Padding the sequences
max_sequence_length = 100  # define your own maximum sequence length
train_padded_sequences = pad_sequences(train_sequences, maxlen=max_sequence_length)
test_padded_sequences = pad_sequences(test_sequences, maxlen=max_sequence_length)

# Splitting the data into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(train_padded_sequences, train_labels, test_size=0.2, random_state=42)

# Define the Keras model
model2 = Sequential()
model2.add(Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=100, input_length=max_sequence_length))
model2.add(LSTM(units=256))
model2.add(Dropout(0.8))
model2.add(Dense(units=1, activation='sigmoid'))

# Compile the model
model2.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
model2.fit(X_train, y_train, epochs=10, batch_size=128, validation_data=(X_val, y_val))

# Predict on the test data
test_predictions2 = model2.predict(test_padded_sequences)
test_predictions2 = (test_predictions2 > 0.5).astype(int)
```

```
Epoch 1/10
48/48 [==============================] - 57s 1s/step - loss: 0.6482 - accuracy: 0.6194 - val_loss: 0.5368 - val_accuracy: 0.
7689
```

Methods Used

In this NLP project, we employ various methods to tackle the task of predicting target labels based on the words used in textual data. The following methods are utilized:

Logistic Regression: Logistic regression is a widely-used statistical method that can be applied to binary classification problems. In this approach, we use the logistic function to model the relationship between the input features (words) and the target labels (0 or 1). By optimizing the logistic regression model using techniques such as gradient descent, we aim to find the best-fitting parameters that maximize the likelihood of the observed data. Logistic regression offers simplicity, interpretability, and efficiency, making it a popular choice for NLP tasks.

TensorFlow Keras: TensorFlow Keras is a high-level deep learning library that provides a user-friendly interface to build and train neural networks. We leverage the power of deep learning by constructing neural network architectures using Keras. These architectures consist of multiple layers of interconnected neurons, enabling the model to learn complex patterns and representations from the textual data. TensorFlow Keras offers a range of pre-built layers and optimization techniques, allowing us to experiment with different network architectures and hyperparameters.

TensorFlow using Sequential Model: Sequential models in TensorFlow allow us to build neural networks with a linear stack of layers. We can create a sequential model by adding layers one after another, specifying the number of neurons, activation functions, and other relevant parameters for each layer. This approach is particularly suitable for NLP tasks as it provides a straightforward way to design and train deep learning models. By utilizing TensorFlow's extensive ecosystem, we can employ advanced techniques like regularization, dropout, and batch normalization to enhance the model's performance and prevent overfitting.

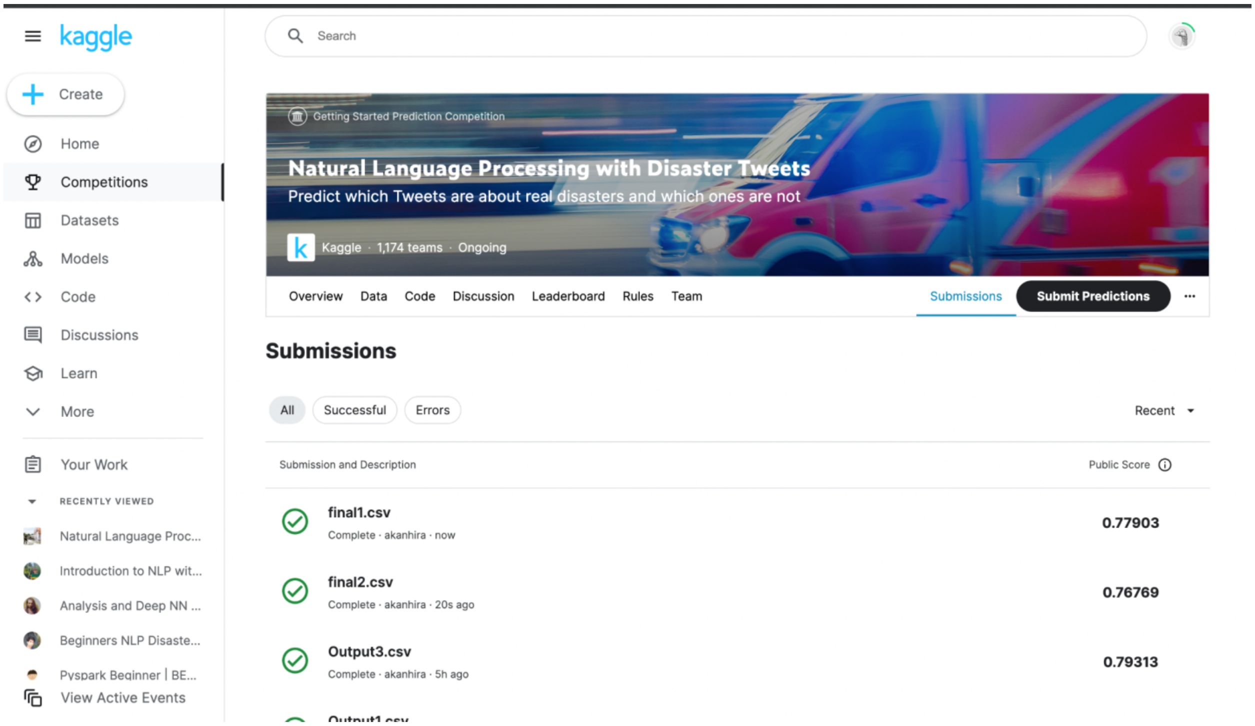
By utilizing logistic regression, TensorFlow Keras, TensorFlow using Sequential Model, and neural networks, we aim to explore a range of methods and algorithms to predict target labels based on the words used in the textual data. These approaches provide us with the means to leverage both traditional statistical techniques and state-of-the-art deep learning models to tackle the task at hand.

Results

Note : One of our submissions recorded a score of 0.79, however we realized that there was an error in that hence we are not considering that as our highest score.

Our Final Kaggle Score is: 0.77903

Image:



Natural Language Processing with Disaster Tweets

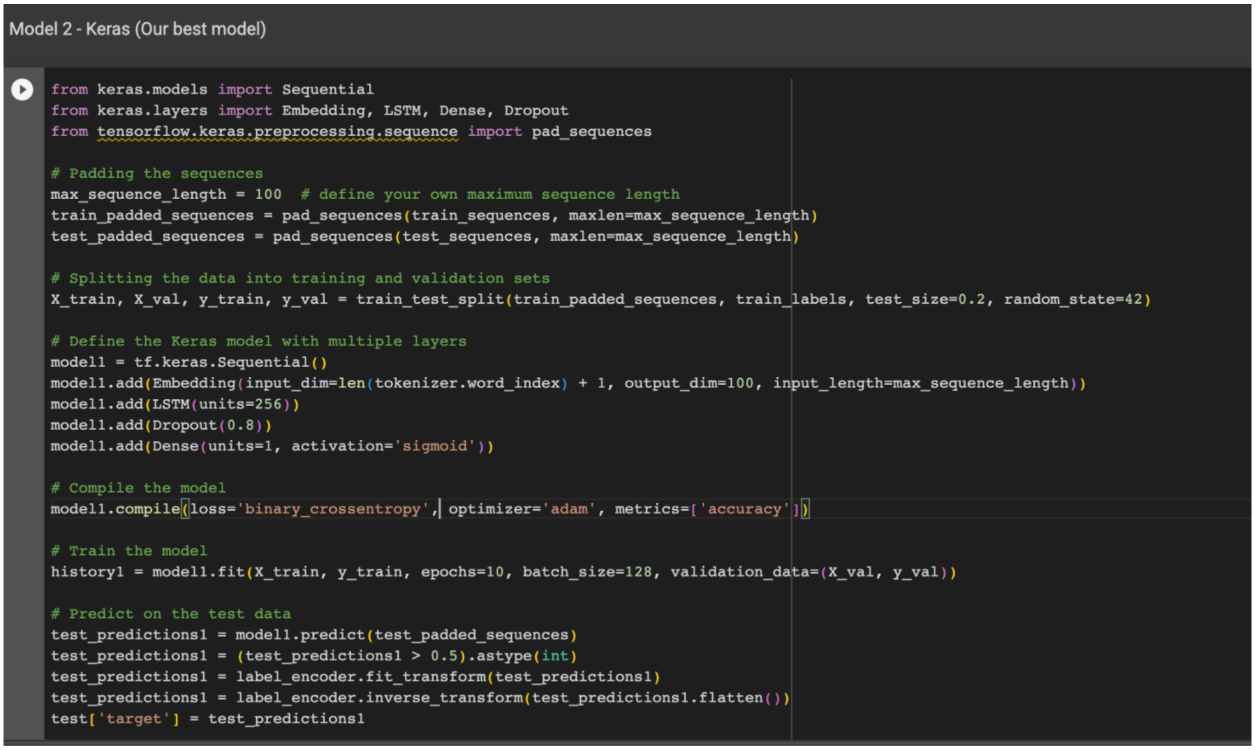
Predict which Tweets are about real disasters and which ones are not

Submissions

Submission and Description	Public Score
final1.csv Complete · akanhira · now	0.77903
final2.csv Complete · akanhira · 20s ago	0.76769
Output3.csv Complete · akanhira · 5h ago	0.79313
Output1.csv	

Although we ran multiple models and uploaded various target file on Kaggle, our final score for the model we've selected is: **0.77903**

The Final Model:



Model 2 - Keras (Our best model)

```
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense, Dropout
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Padding the sequences
max_sequence_length = 100  # define your own maximum sequence length
train_padded_sequences = pad_sequences(train_sequences, maxlen=max_sequence_length)
test_padded_sequences = pad_sequences(test_sequences, maxlen=max_sequence_length)

# Splitting the data into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(train_padded_sequences, train_labels, test_size=0.2, random_state=42)

# Define the Keras model with multiple layers
model1 = tf.keras.Sequential()
model1.add(Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=100, input_length=max_sequence_length))
model1.add(LSTM(units=256))
model1.add(Dropout(0.8))
model1.add(Dense(units=1, activation='sigmoid'))

# Compile the model
model1.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
history1 = model1.fit(X_train, y_train, epochs=10, batch_size=128, validation_data=(X_val, y_val))

# Predict on the test data
test_predictions1 = model1.predict(test_padded_sequences)
test_predictions1 = (test_predictions1 > 0.5).astype(int)
test_predictions1 = label_encoder.fit_transform(test_predictions1)
test_predictions1 = label_encoder.inverse_transform(test_predictions1.flatten())
test['target'] = test_predictions1
```

Work done by each team member

Initial EDA and Data Cleaning: Shrushti Shah and Anirudh Murali
Model Building: Arvind Kanhirampara Ravi, Charishma Jaladi, Anirudh Murali and Madison Sanchez
Documentation: Shrushti Shah and Madison Sanchez

Conclusion and Learnings

In conclusion, this NLP project focuses on predicting target labels based on the words used in textual data. We have explored and applied various methods, including logistic regression, TensorFlow Keras, TensorFlow using Sequential Model, and neural networks. These methods provide us with a diverse range of approaches to tackle the task of text classification.

Through the use of logistic regression, we leverage statistical modeling techniques to capture the relationship between input features and target labels, providing a simple and interpretable solution. TensorFlow Keras allows us to harness the power of deep learning, constructing neural network architectures that can learn complex patterns and representations from the textual data.

By utilizing TensorFlow using Sequential Model, we can design linear stack models efficiently, taking advantage of TensorFlow's extensive ecosystem. Neural networks, inspired by the human brain, enable us to capture semantic and contextual information from text, allowing for sophisticated feature learning and prediction.

By employing these methods, we aim to build models that can accurately predict the target labels based on the words used in the given textual data. These models have the potential to provide valuable insights and aid decision-making processes in various applications such as sentiment analysis, spam detection, and fake news identification.

Throughout the project, we have also addressed challenges specific to NLP tasks, including handling noisy or unstructured text, addressing class imbalance issues, and mitigating bias in the data. These considerations contribute to the robustness and reliability of our models.

In conclusion, this NLP project serves as a stepping stone towards advancing text classification techniques. The exploration and application of various methods have provided us with valuable insights into the predictive capabilities of Logistic Regression, TensorFlow Keras, TensorFlow using Sequential Model. By leveraging these methods, we can contribute to the development of more accurate and efficient NLP systems, opening up possibilities for improved decision-making and analysis in the realm of textual data.

Learnings:

Shrushti Shah

I have learnt that the quality and cleanliness of the training data have a significant impact on the performance of NLP models. Proper preprocessing techniques, such as tokenization, stemming, lemmatization, and removing noise or irrelevant information, are crucial to improve the accuracy and efficiency of models. Also choosing the appropriate model architecture and fine-tuning its hyperparameters can significantly affect the performance of NLP systems. Experimentation with different models and parameter settings is often necessary to achieve optimal results.

Madison Sanchez

From this project, I have gained a deeper understanding about how computers process and understand human language. Also, I have learned that language models require a lot of preparation and that experimentation with different parameters is crucial when trying to achieve optimal results. Further, data preprocessing techniques such as stemming, removal of stop words, lemmatization are extremely important when preparing the data and help to improve the model accuracy significantly. All in all, this project permitted me the opportunity to experiment and learn more about modeling techniques.

Arvind Kanhirampara Ravi

By building and experimenting with multiple models, I’ve understood the real function of epochs and how it is related to overfitting. Training the model iteratively by increasing the number of epochs can help improve accuracy but can also risk overfit. Hence, finding the right number of epochs to set can be crucial while building models. Lastly I also gained insight into the relationship between validation accuracy and train accuracy. As the number of epoch increases, as expected the training accuracy will also increase, however the training accuracy will relatively stay constant. If the validation accuracy continues to decrease as training accuracy increases, this could be another indication of overfit.

Charishma Jaladi

I have gained valuable insights into various aspects of machine learning and natural language processing. I learned about Tokenization as effective methods to convert text data into numerical features. Additionally, I discovered the concept of padded sequences, which involves ensuring that all input sequences have the same length by adding padding or truncating as necessary. This ensures compatibility in model training and improves computational efficiency. Logistic Regression and keras models, along with the concept of embedding layers.

I also discovered the importance of model evaluation and metrics like accuracy score.

Overall, these learnings have equipped me with a solid foundation in machine learning techniques and their practical application in text analysis tasks.

Anirudh Murali

Through my involvement in this project, I gained valuable experience in both data cleaning and model building. I recognized the significance of clean and high-quality data in achieving accurate and reliable models. Data cleaning allowed me to address issues such as missing values, outliers, and inconsistencies, which greatly influenced the performance of the models. Furthermore, I acquired knowledge and skills in building regression models and TensorFlow models. Regression models helped me understand and predict relationships between variables, while TensorFlow models provided me with a deeper understanding of neural networks and their applications in machine learning.

An essential lesson I learned was the importance of continuous improvement in model accuracy. I realized that achieving higher accuracy requires a thorough understanding of the code and model behavior. By delving into the code, I gained insights into how changes in parameters and configurations impact the model's performance. This comprehension empowered me to fine-tune the models effectively, optimizing their accuracy and enhancing their overall performance.

Overall, this project has significantly enhanced my proficiency in using Python for modeling tasks. It has not only strengthened my technical skills but also instilled in me the confidence to undertake similar projects in the future. I now feel more comfortable exploring and leveraging Python's capabilities to develop advanced models and extract valuable insights from data.