Mental Health Report

CS4662 Machine Learning

Team:
Andres Dominguez
Pierce Nance
Kaelyn Taing

Table of Contents:

Introduction

Mental health is a critical concern in modern society, with depression being one of the most prevalent and challenging conditions affecting individuals worldwide. This project leverages machine learning algorithms to analyze the Depression Survey/Dataset for Analysis from Kaggle, aiming to identify patterns and predictors that correlate with a risk of depression. By understanding these contributing factors, we hope to support preventative strategies and promote mental wellness using data-driven insights.

Project Description

The dataset used in this study was derived from an anonymous survey conducted between January and June 2023 across various cities, targeting adults aged 18 to 60 from diverse professional and educational backgrounds. Participants voluntarily provided inputs on several demographic and lifestyle-related factors without undergoing clinical evaluations.

Notably, the dataset we worked on was generated using a deep learning model trained on the original Depression Survey dataset. While the distributions are close to the original data, they are not identical. The target variable, **Depression**, is a binary classification indicating whether an individual is at risk of depression (`Yes` or `No`), based on their survey responses.

We employed several machine learning models for predictive analysis:

- Logistic Regression
- Random Forest
- Artificial Neural Network (ANN)
- CatBoost

This topic holds significant social relevance as the growing burden of mental health challenges underscores the need for early detection and intervention. Our project investigates how machine learning can aid in identifying high-risk individuals and uncovering underlying trends in depression-related factors.

Goals: Our goal for this project was to use and compare multiple types of models to determine which would be most suitable for our particular dataset.

View of the Data

The dataset includes the following features:

- Demographic/Contextual: Gender, Age, City, Working Professional or Student, Profession, Degree
- Lifestyle & Behavioral: Academic Pressure, Work Pressure, Study Satisfaction, Job Satisfaction, Sleep Duration, Dietary Habits, Work/Study Hours
- Psychological & Historical: Financial Stress, Have you ever had suicidal thoughts?, Family History of Mental Illness
- Academic: CGPA

	id	Name	Gender	Age	City	Working Professional or Student	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	Have you ever had suicidal thoughts ?	Work/ Study Hours	Financial Stress	Family History of Mental Illness	Depression
0	0	Aaradhya	Female	49.0	Ludhiana	Working Professional	Chef	0.0	5.0	0.00	0.0	2.0	More than 8 hours	Healthy	BHM	No	1.0	2.0	No	0
1	1	Vivan	Male	26.0	Varanasi	Working Professional	Teacher	0.0	4.0	0.00	0.0	3.0	Less than 5 hours	Unhealthy	LLB	Yes	7.0	3.0	No	1
2	2	Yuvraj	Male	33.0	Visakhapatnam	Student	None	5.0	0.0	8.97	2.0	0.0	5-6 hours	Healthy	B.Pharm	Yes	3.0	1.0	No	1
3	3	Yuvraj	Male	22.0	Mumbai	Working Professional	Teacher	0.0	5.0	0.00	0.0	1.0	Less than 5 hours	Moderate	BBA	Yes	10.0	1.0	Yes	1
4	4	Rhea	Female	30.0	Kanpur	Working Professional	Business Analyst	0.0	1.0	0.00	0.0	1.0	5-6 hours	Unhealthy	BBA	Yes	9.0	4.0	Yes	0

Numerical Features	Categorical Features
Age	Gender
Academic Pressure	City
Work Pressure	Working Professional or Student
CGPA	Profession
Study Satisfaction	Sleep Duration
Job Satisfaction	Dietary Habits
Work/Study Hours	Degree
Financial Stress	Have you ever had suicidal thoughts ?
	Family History of Mental Illness

```
# Define Categorical Columns
cat_cols = ["Gender", "City", "Working Professional or Student",
            "Profession", "Sleep Duration", "Dietary Habits", "Degree",
            "Have you ever had suicidal thoughts ?", "Family History of Mental Illness"
]

# Define Numerical Columns
num_cols = ["Age", "Academic Pressure", "Work Pressure", "CGPA",
            "Study Satisfaction", "Job Satisfaction",
            "Work/Study Hours", "Financial Stress"]
```

Preprocessing Data

To properly train most of our machine learning models, several transformations to the data needed to made:

```
# Setting Empty Numerical Columns to 0
dataset[num_cols] = dataset[num_cols].fillna(0)

# set so there's no NaN values
dataset[cat_cols] = dataset[cat_cols].fillna("None").astype(str)

# Define Features
features = dataset.columns.tolist()
features.remove('id')
features.remove('Name')
features.remove('Depression')

# Define Label
label = dataset['Depression']

# define X and y
X = dataset[features]
y = dataset['Depression']

# One Hot Encoding
X_encoded = pd.get_dummies(X, columns=cat_cols, drop_first=True)
y_encoded = pd.get_dummies(y, columns=['Depression'], drop_first=True).values.ravel()

dataset.head()
```

- **Column Removal**: Irrelevant fields such as `id` and `Name` were dropped from the dataset as they do not contribute to predictive modeling. They also interfere with proper One Hot Encoding of training and test sets.
- **Missing Values**:
    - Numerical features with missing values were imputed with 0.
    - Categorical features with missing values were imputed with None (string-based placeholder).
- **Encoding**:
    - One-hot encoding was applied to categorical features to prepare the data for models that require numerical inputs.
    - **CatBoost** was treated differently, as it natively supports categorical data and does not require explicit encoding.

Models

1. Logistic Regression

A supervised learning model that predicts the probability that an input belongs to a particular class, especially in binary classification.

```
#Splitting on Encoded Data
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y_encoded, test_size=0.2, random_state=2)

# Intialize Logistic Regression with max_iter at 1000
logreg = LogisticRegression(solver='lbfgs', max_iter=1000)

# Training and Testing
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
```

Logistic Regression is efficient and fast to train and scales well to moderately large datasets. It is also much less prone to overfitting compared to other models because of its simplicity, as it defines a linear decision boundary compared to the complex decision boundaries of models like neural networks, making it less likely to capture noise and irrelevant patterns in the training data. As Logistic Regression is widely used for binary classification tasks, our output of either 0 (negative) or 1 (positive) for depression is well-suited for this model.

2. Random Forest Algorithm

A meta estimator that performs ensemble learning through constructing numerous decision tree classifiers.

```
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y_encoded, test_size=0.2, random_state=2)

#Initialize Random Forest Model
rf = RandomForestClassifier(n_estimators = 29, bootstrap = True, random_state=2)

# Training and Testing
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```

Random forest uses averaging between decision tree classifiers to improve the predictive accuracy and to control over-fitting. It also scales well with large, complex data without significant performance degradation. Since random forest is versatile and can be applied to classification tasks in general, we decided to use it for our dataset.

3. Artificial Neural Network (ANN)

A neural network model inspired by the structure and function of the human brain and make predictions.

```
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y_encoded,
                                                    test_size=0.2,
                                                    random_state=2)

#Initialize ANN multi layer perceptron model
ann = MLPClassifier(hidden_layer_sizes=(5,5), activation= 'logistic',
                        solver='adam', alpha=1e-5, random_state=1,
                        learning_rate_init = 0.1, verbose=True, tol=0.0001)

# Training and Testing
ann.fit(X_train, y_train)
y_pred = ann.predict(X_test)
```

ANN can scale effectively with increased data and computing power, and specializes in working with unstructured, high-dimensional data. NNs excel at capturing complex, non-linear relationships in data through activation functions. As our dataset was very large, we decided to use ANN as one of our models.

4. CatBoost

A model implementing Ensemble Learning, is similar to XGBoost/ADABoost

```
# Split the Dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42)
# Pooling Data
train_pool = Pool(X_train, y_train, cat_features=cat_cols)
test_pool = Pool(X_test, y_test, cat_features=cat_cols)

# Create Catboost Model
catboost = CatBoostClassifier(verbose=0)

#Training and Testing
catboost.fit(train_pool)
y_pred = catboost.predict(X_test)
```

CatBoost requires little data processing and has high accuracy and speed on structured data. It also interprets categorical features natively. As our dataset had many categorical features and data points, we decided to use this model.

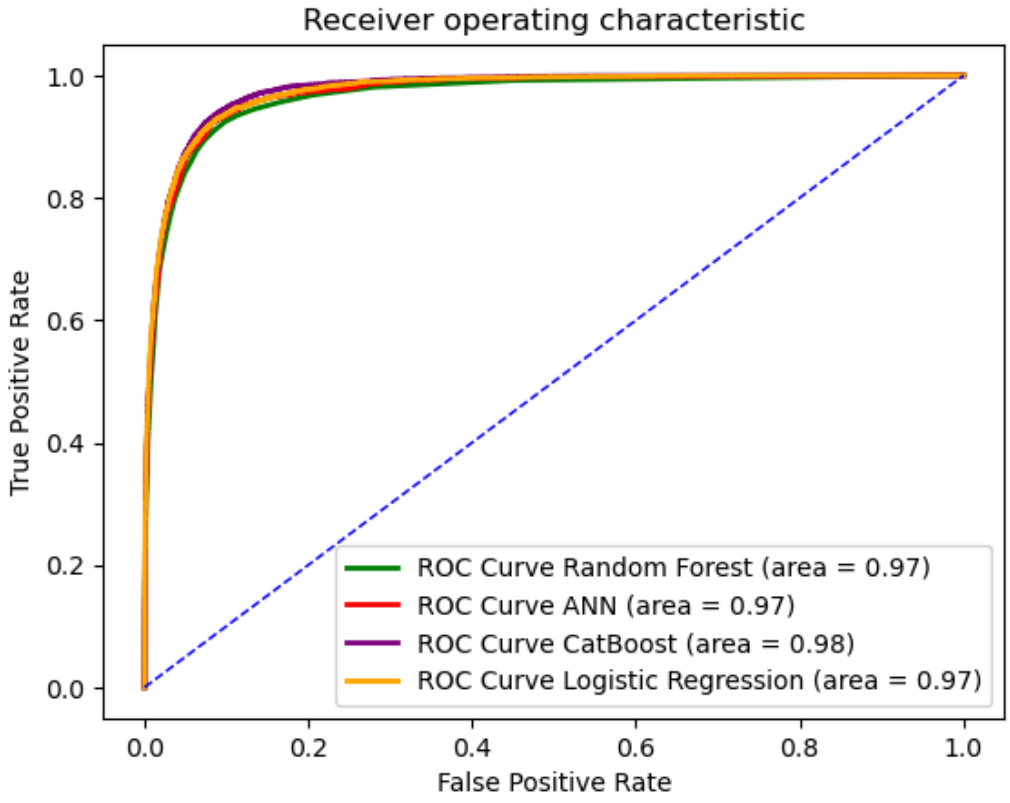
Results and Comparison

Accuracy Scores (for each Model)

Models	Accuracy
CatBoost	0.938272921108742
Logistic Regression	0.9384150675195452
ANN	0.932089552238806
Random Forest	0.930525941799716

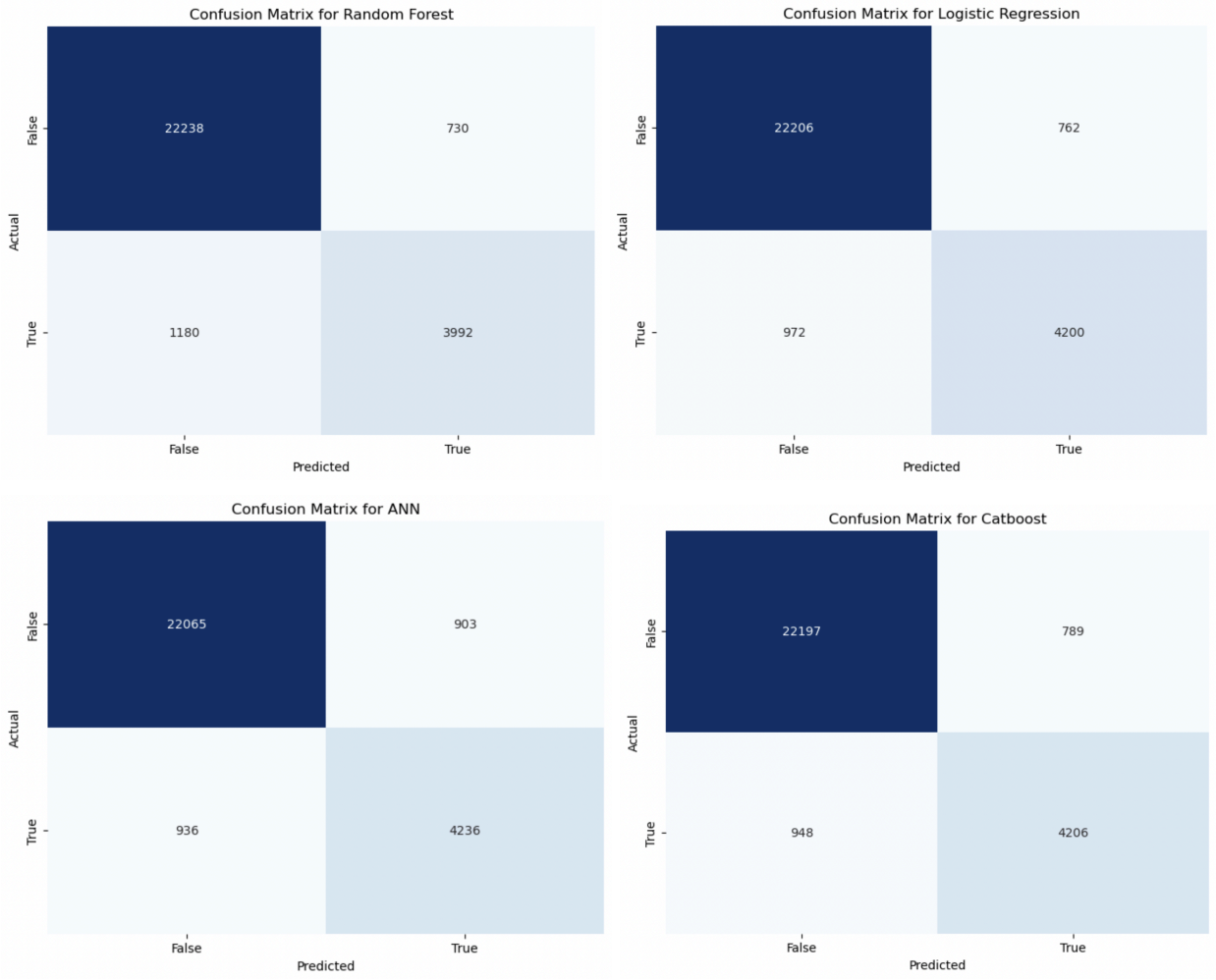
While Logistic Regression reached the highest accuracy out of all the models, each model produced relatively similar scores, landing around 93 - 94% accuracy.

ROC Curve Results (for each Model)



Catboost appears to have the highest AUC with 0.98, but all models produced relatively similar scores, landing around a 0.97-0.98 AUC.

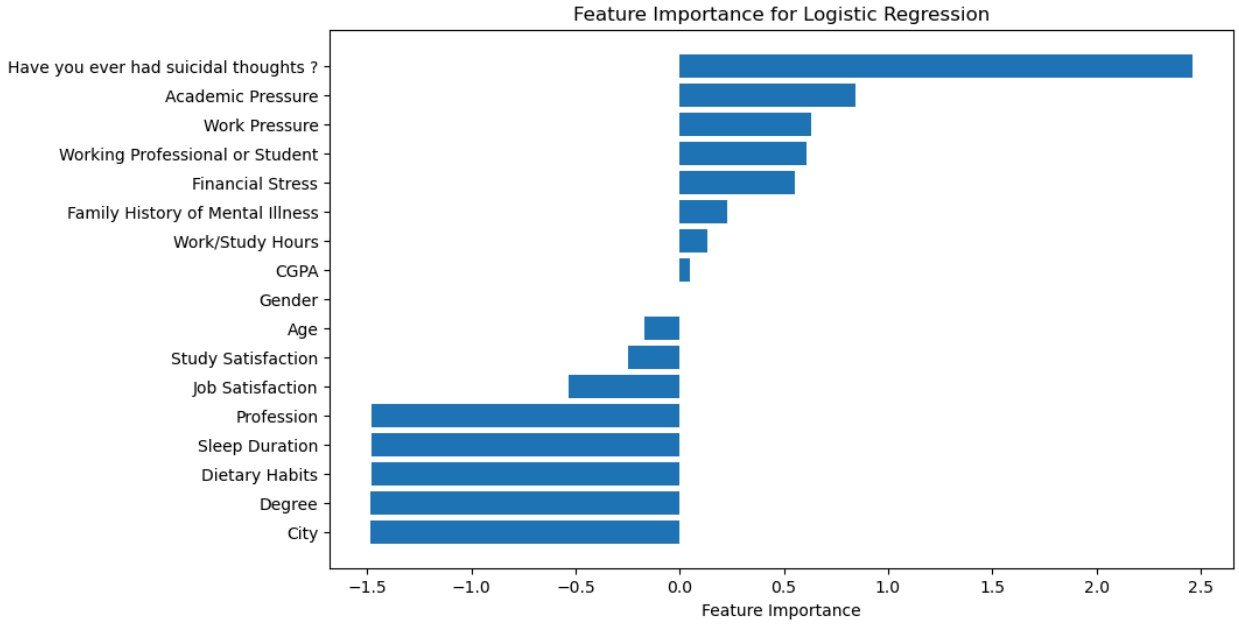
Confusion Matrices (for each Model)

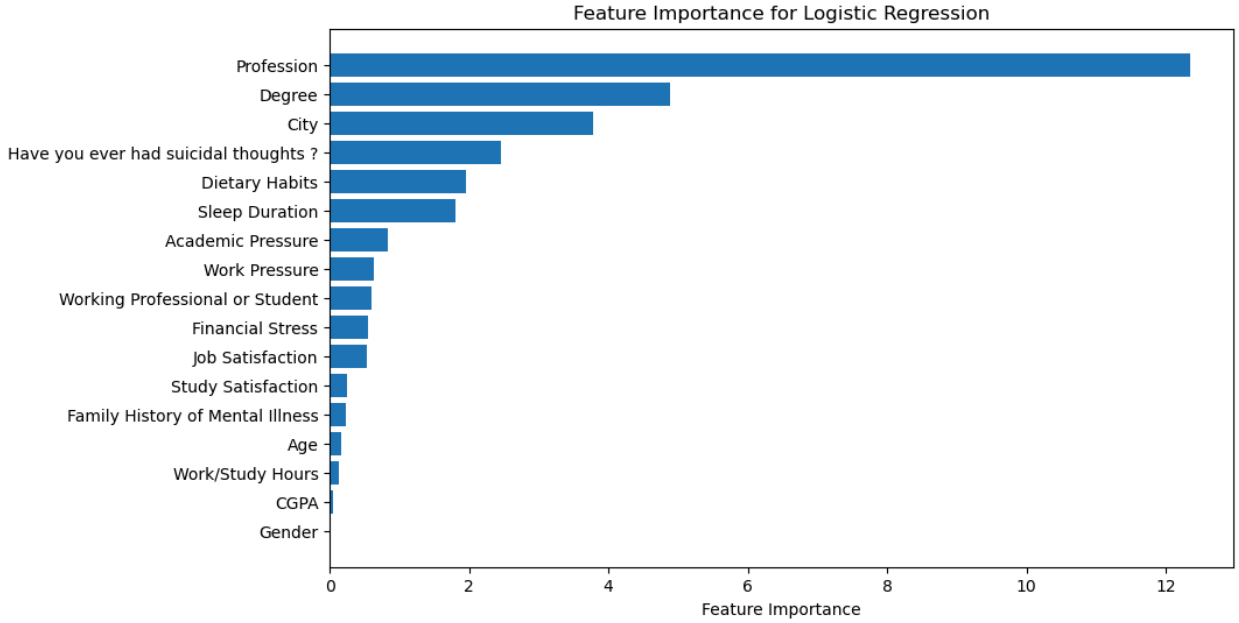


Confusion Matrices help us see the prediction summary in matrix form. True Negatives share extremely similar results, with each value in the 22000's. However we see more variance within the True Positives, with values between 3700 and 4200.

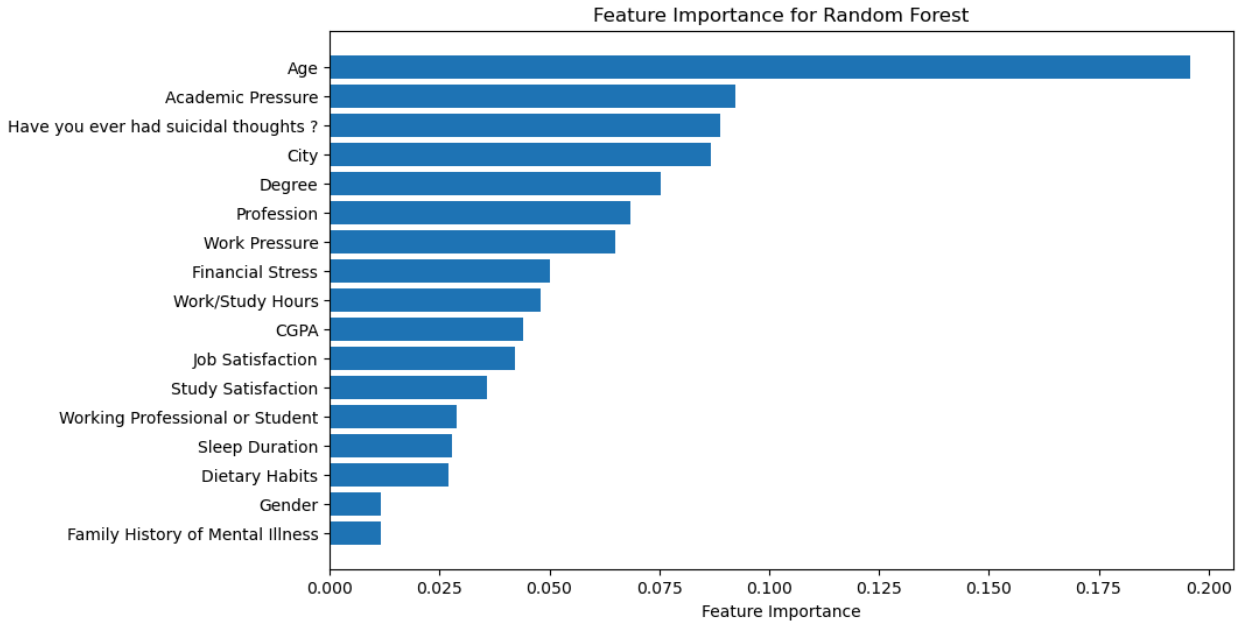
Feature Importance Graphs (for each Model)

Logistic Regression:

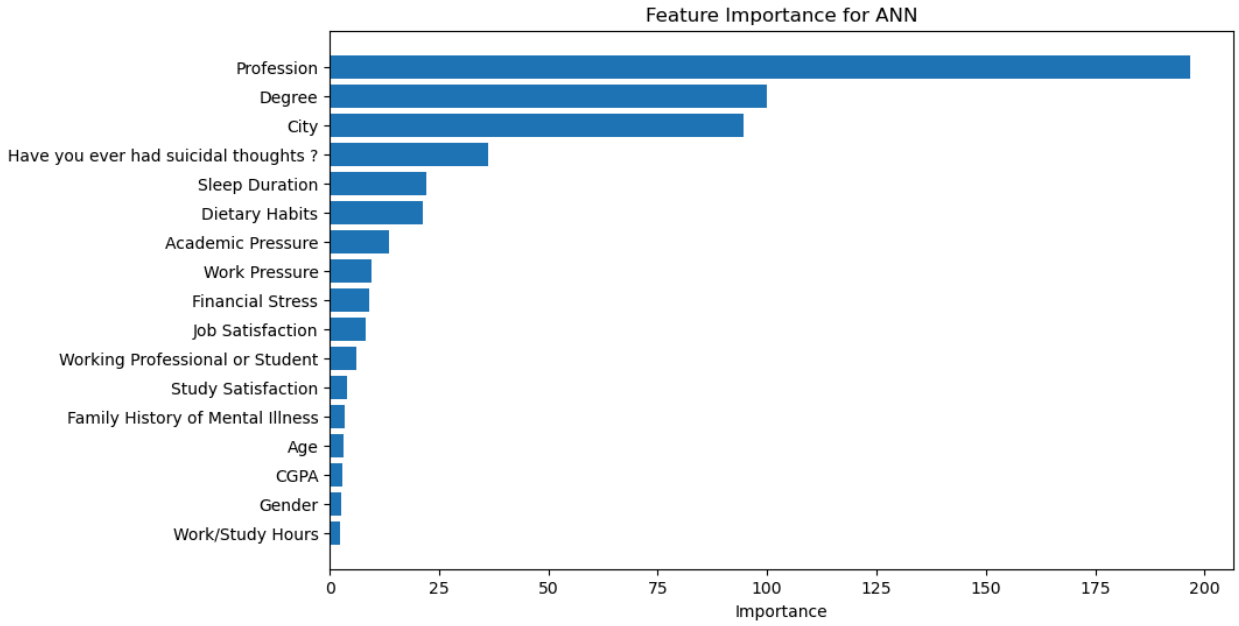


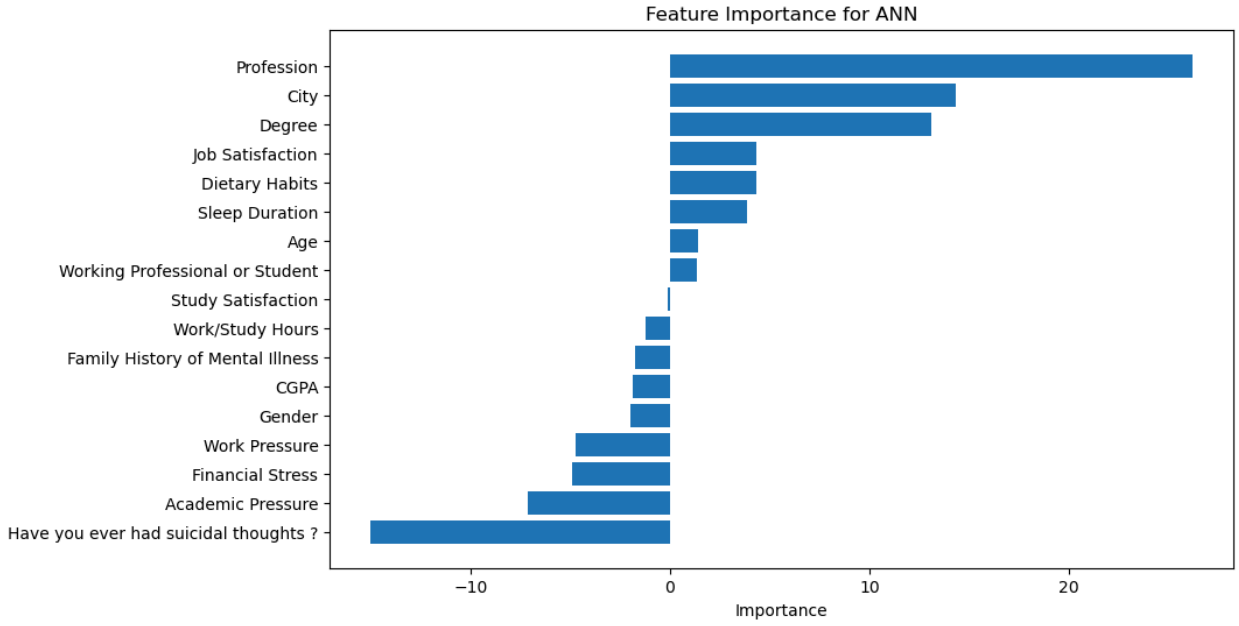


Random Forest:

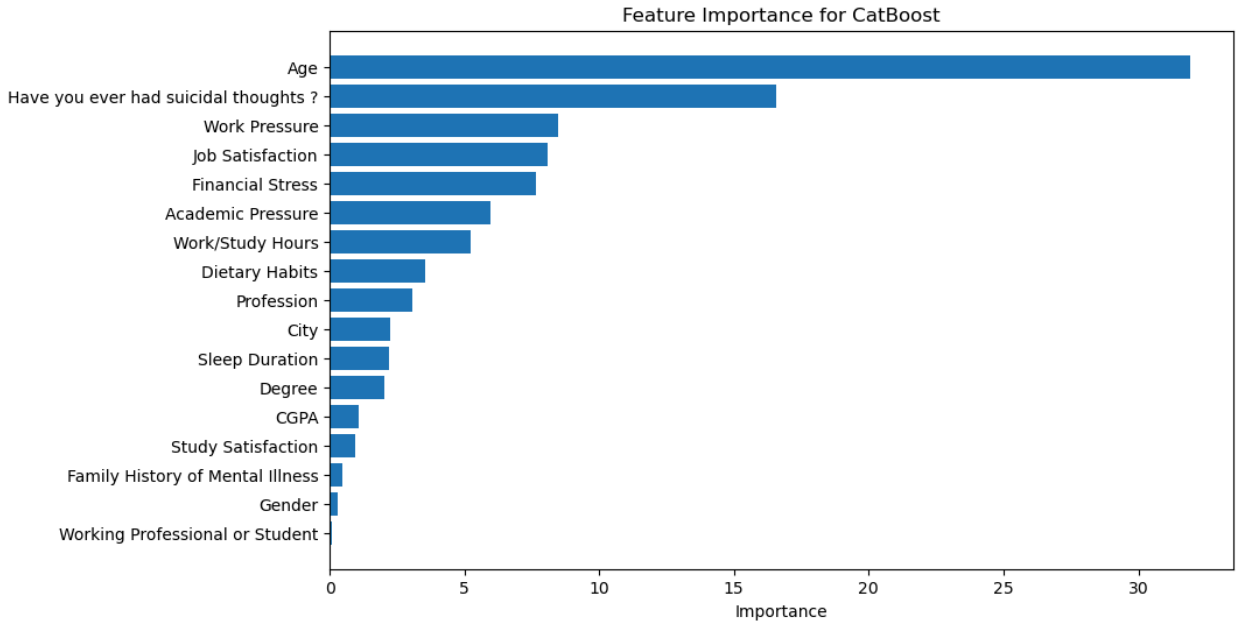


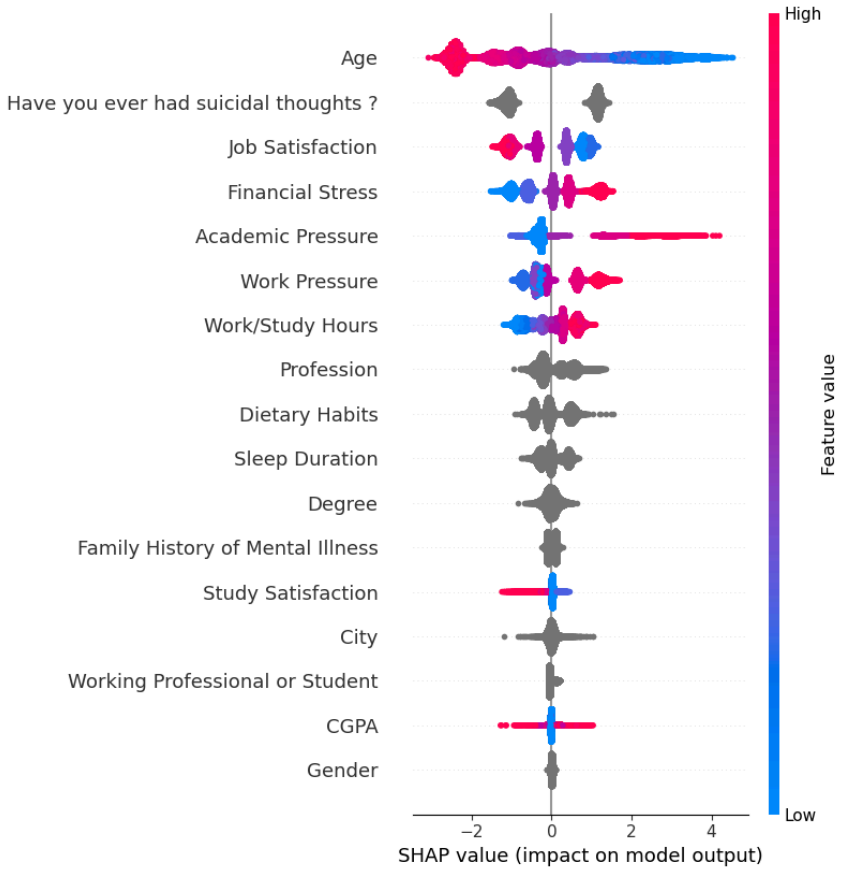
ANN:


Feature Importance for ANN
Profession
Degree
City
Have you ever had suicidal thoughts ?
Sleep Duration
Dietary Habits
Academic Pressure
Work Pressure
Financial Stress
Job Satisfaction
Working Professional or Student
Study Satisfaction
Family History of Mental Illness
Age
CGPA
Gender
Work/Study Hours
0
25
50
75
100
125
150
175
200
Importance


Feature Importance for ANN
Profession
City
Degree
Job Satisfaction
Dietary Habits
Sleep Duration
Age
Working Professional or Student
Study Satisfaction
Work/Study Hours
Family History of Mental Illness
CGPA
Gender
Work Pressure
Financial Stress
Academic Pressure
Have you ever had suicidal thoughts ?
−10
0
10
20
Importance

Catboost:


Feature Importance for CatBoost
Age
Have you ever had suicidal thoughts ?
Work Pressure
Job Satisfaction
Financial Stress
Academic Pressure
Work/Study Hours
Dietary Habits
Profession
City
Sleep Duration
Degree
CGPA
Study Satisfaction
Family History of Mental Illness
Gender
Working Professional or Student
0
5
10
15
20
25
30
Importance


Age
Have you ever had suicidal thoughts ?
Job Satisfaction
Financial Stress
Academic Pressure
Work Pressure
Work/Study Hours
Profession
Dietary Habits
Sleep Duration
Degree
Family History of Mental Illness
Study Satisfaction
City
Working Professional or Student
CGPA
Gender
−2
0
2
4
SHAP value (impact on model output)
High
Low
Feature value

Throughout the feature importance graphs we see that each feature is displayed with varying degrees of impact. It's been found that Age, City and Degree plays a bigger factor in whether someone has depression compared to others. But the question of "Have you ever had suicidal thoughts?" usually comes in close second, with Academic Pressure, Profession, and City following close behind. Logistic Regression and ANN are are based on coefficients the models figure out during training, so negative values for the features can be displayed (and can be completely positive by wrapping coefficients in absolute value). Logistic Regression and CatBoost have feature importance functions built in, and were therefore used for graphing.

Conclusion

We can draw from this project that the four models are accurate and produce good but slightly varied results. We see that Logistic Regression has a high accuracy score, but produces interesting graph dispersion with its use of coefficients, making "City" and "Have you ever had suicidal thoughts?" the big values. Random Forest is also accurate, with its algorithm determining "Age" as a massive factor when it comes to depression. ANN scales the data effectively with a good score as well and makes "Profession" the biggest factor in depression. CatBoost covers its bases with the categorical data very well, producing the best accuracy score/ROC curve. Additionally, CatBoost also finds that "Age" plays a big role in depression much like Random Forest. From these findings, we can determine that the four biggest causes of depression are Age, City, Profession, and "Have you ever had suicidal thoughts?", which make sense given the certain circumstances surrounding each feature. Logistic Regression performed with the highest accuracy score. CatBoost performed with a higher AUC and better results for the confusion matrix overall.

Contributions/Responsibilities

Andres Dominguez: Preprocessing Data (One Hot Encoding), ANN
Pierce Nance: Random Forest, Logistic Regression
Kaelyn Taing: Preprocessing Data (Numerical and Categorical Data Processing and Filtering), CatBoost, Logistic Regression
Everyone: Accuracy Score, ROC, Confusion Matrix, Feature Importance (For respective models)