AI Project 2

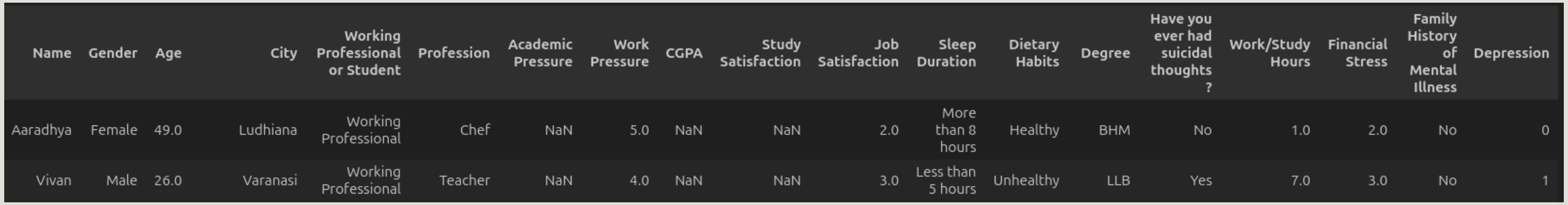
FINAL PRESENTATION

Group_A2_13
Sérgio Rego Nossa (up202206856)
Pedro Marinho (up202206854)
Xavier Martins (up202206632)

Problem Definition

We were tasked to create an AI model to predict depression based on common risk factors / symptoms (sleep duration, work / academic pressure, etc.)

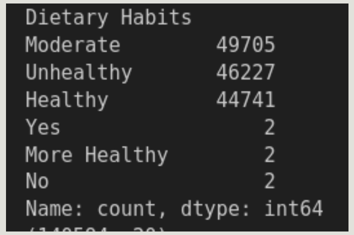
The provided dataset includes individual records, with each row containing relevant information to assess depression, along with a label indicating whether the person is experiencing depression.



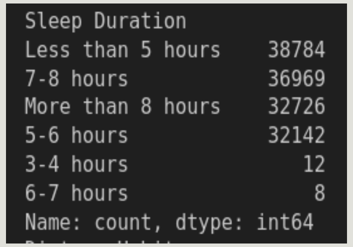
Name	Gender	Age	City	Working Professional or Student	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	Have you ever had suicidal thoughts ?	Work/Study Hours	Financial Stress	Family History of Mental Illness	Depression
Aaradhya	Female	49.0	Ludhiana	Working Professional	Chef	NaN	5.0	NaN	NaN	2.0	More than 8 hours	Healthy	BHM	No	1.0	2.0	No	0
Vivan	Male	26.0	Varanasi	Working Professional	Teacher	NaN	4.0	NaN	NaN	3.0	Less than 5 hours	Unhealthy	LLB	Yes	7.0	3.0	No	1

Preprocessing

As for the process of preprocessing our dataset we have passed by the following steps:



```
Dietary Habits
Moderate         49705
Unhealthy        46227
Healthy          44741
Yes                  2
More Healthy         2
No                   2
Name: count, dtype: int64
```



```
Sleep Duration
Less than 5 hours    38784
7-8 hours            36969
More than 8 hours    32726
5-6 hours            32142
3-4 hours               12
6-7 hours                8
Name: count, dtype: int64
```

Cleaning and modifying columns

At this step we refined certain categorical features to reduce noise and improve model clarity. For example:

In the **Sleep Duration** column, we retained only the top 4 most frequent values.

In **Dietary Habits**, we kept the top 3 most representative categories.

These adjustments help prevent the model from being influenced by infrequent or ambiguous values.

To simplify the dataset and address sparsity due to missing values, we merged related columns:

Job Satisfaction and **Study Satisfaction** were combined into a single **Satisfaction** column.

Work Pressure and **Academic Pressure** were merged into a unified **Pressure** column.

These changes allow us to treat students and professionals under the same feature, reducing dimensionality and easing the modeling process.

For cases of missing data, we have done the following:

For **numerical columns**, we filled missing values using the **median**. For example, in **CGPA**, which is only applicable to students, we imputed a value of **–1** for professionals.

For **categorical columns**, we used the **mode** to replace missing values, ensuring the most common category is used.

Preprocessing

Split dataset

At this phase we separated the target variable **Depression** from feature set and split our current dataset in a train and test set.

Encoding

In the encoding phase we have handled the categorical values, by the following:

Binary Variable were mapped to 0/1 to allow direct numerical interpretation

Ordinal Variables were mapped based on their logic order (Ex. Dietary Habits – unhealthy < moderate < healthy)

Nominal Variables were hot-encoded to represent each category as a separate feature, avoiding any implied order.

Normalization

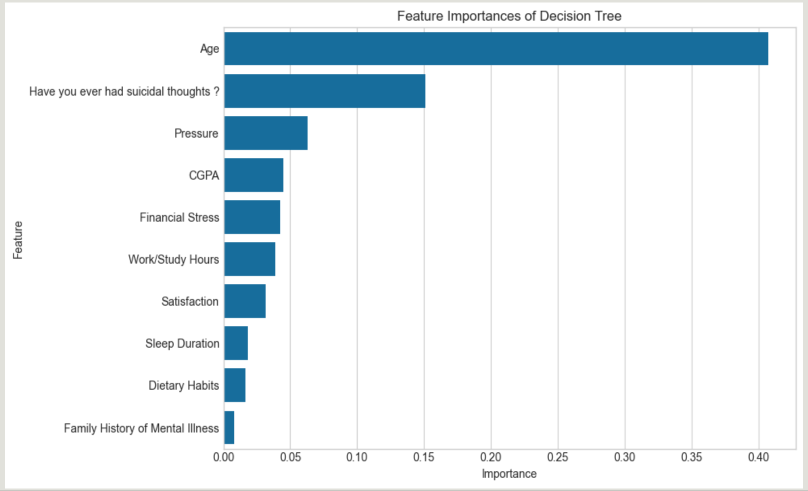
We also normalized the data since the EDA phase revealed that features have different scales (e.g., CGPA vs binary values). To ensure that no feature dominates due to its magnitude, we applied standardization using `StandardScaler`.

Models Implemented

The models that we have implemented were the ones recommended in the assignment document, namely **Decision Tree**, **k-Nearest Neighbors** and **SVM**. Additionally, we included more advanced ensemble and gradient boosting models such as **LightGBM**, **CatBoost**, and **Random Forest** to enhance predictive performance.

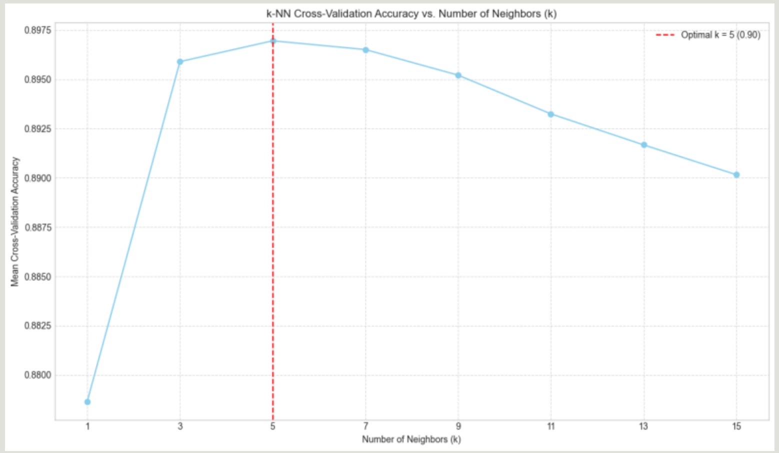
Decision Tree

A tree-based model that splits the data using decision rules inferred from the features. It's interpretable and handles both categorical and numerical data. To this model we concluded that the most influential feature was **age.**



k-Nearest Neighbors(k-NN)

An instance-based model that predicts the label based on the majority class among the k-nearest training samples in the feature space. For this model, we tested different values of *k* to determine the optimal number of neighbors. We found that **k = 5** yielded the best performance.

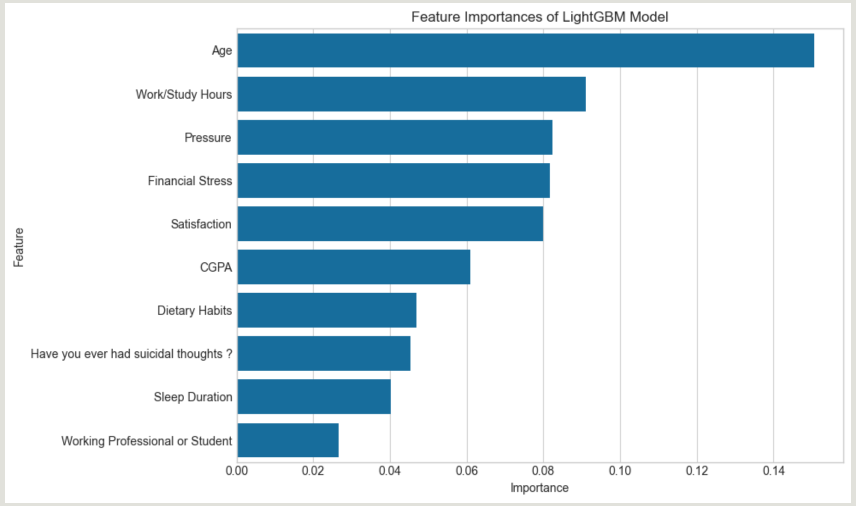


Models Implemented

LightGBM (Light Gradient Boosting Machine)

A fast, distributed, high-performance gradient boosting framework based on decision tree algorithms. It's optimized for speed and efficiency and performs well on large datasets. It works by building many individual decision trees to learn from and correct the errors of previous trees.

In terms of most impactful feature, as well as the Decision Tree, it makes sense for the **age** to be the top 1, because both use Decision Tree's, for the selection of the best route, however the LightGBM, uses not just one but many of them. For that reason, while still sharing the most influential feature, the percentage of the importance is different as well as the order of the other features importance.



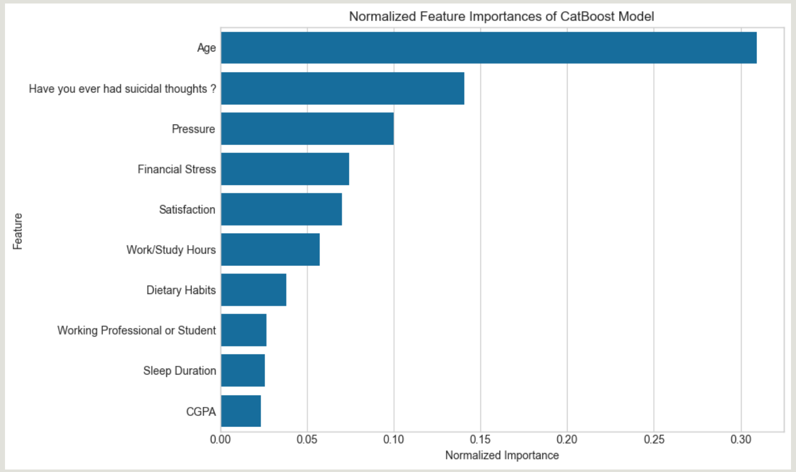
SVM

A model that finds the hyperplane which best separates the classes in a high-dimensional space. We used an RBF kernel for non-linear boundaries. This model can have a better accuracy, however it's very slow compared to the others.

Models Implemented

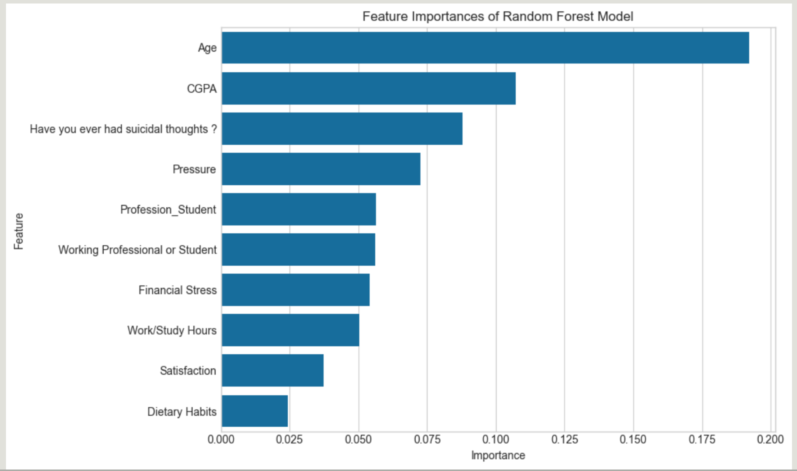
CatBoost

A gradient boosting algorithm, designed to handle categorical features automatically and efficiently, offering good accuracy with less parameter tuning. This is very similar to LightGBM, however can handle categorical features automatically, without the need of manual preprocessing.
As in the previous ones, **age** is also the meaningfull variable.

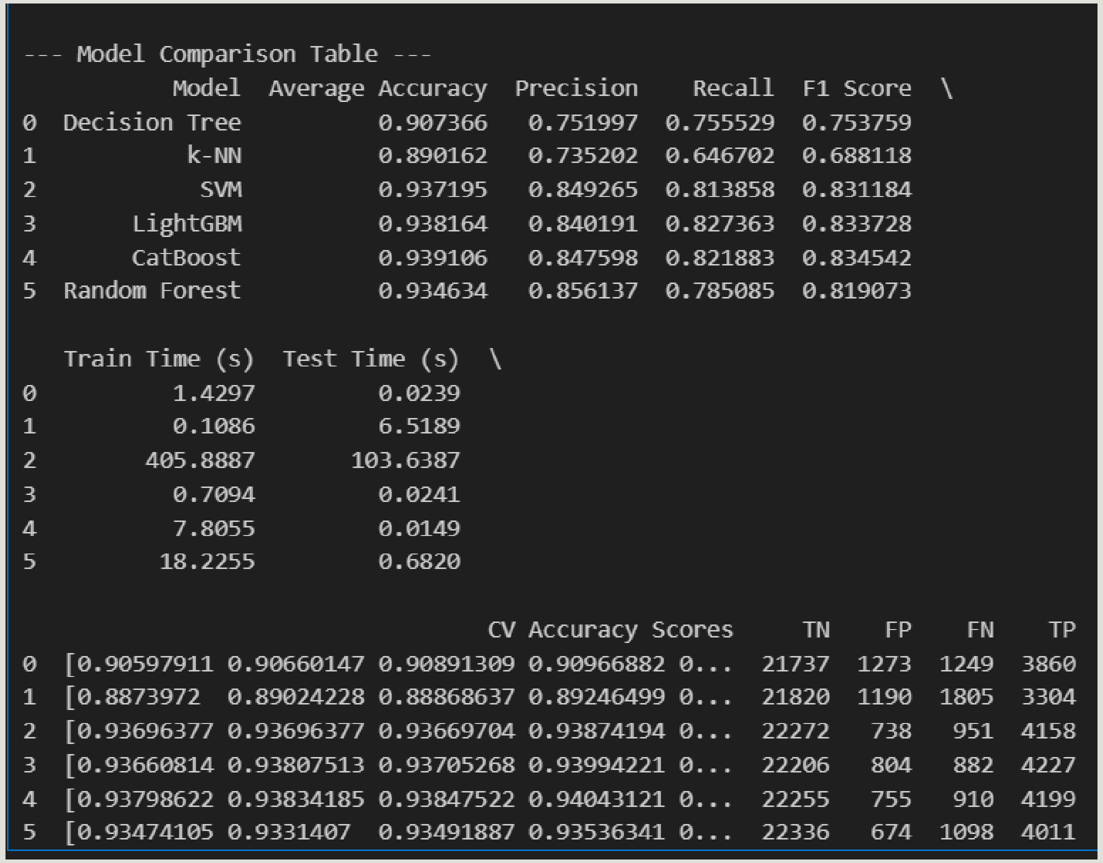


Random Forest

An ensemble learning method that fits multiple decision trees on various sub-samples of the dataset and averages them to improve accuracy and control overfitting. Each tree is trained on a random subset of the data and considers only a random subset of features for splitting, which ensures diversity among the trees.
Once again **age** is the most importante feature.



Evaluation and comparison of models

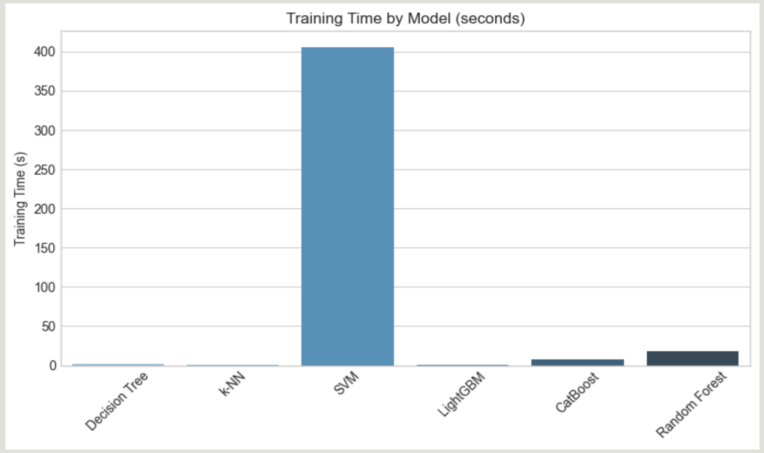


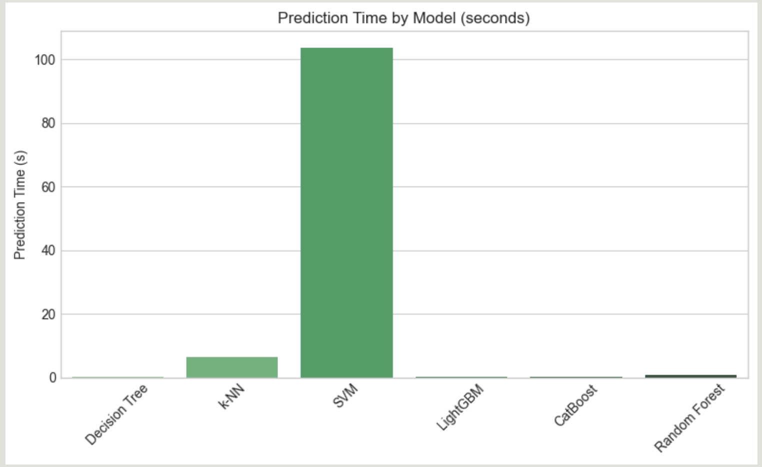
```
--- Model Comparison Table ---
          Model  Average Accuracy  Precision    Recall  F1 Score  \
0  Decision Tree          0.907366   0.751997  0.755529  0.753759
1           k-NN          0.890162   0.735202  0.646702  0.688118
2            SVM          0.937195   0.849265  0.813858  0.831184
3       LightGBM          0.938164   0.840191  0.827363  0.833728
4       CatBoost          0.939106   0.847598  0.821883  0.834542
5  Random Forest          0.934634   0.856137  0.785085  0.819073

   Train Time (s)  Test Time (s)  \
0          1.4297         0.0239
1          0.1086         6.5189
2        405.8887       103.6387
3          0.7094         0.0241
4          7.8055         0.0149
5         18.2255         0.6820

                                  CV Accuracy Scores     TN    FP    FN    TP
0  [0.90597911 0.90660147 0.90891309 0.90966882 0...  21737  1273  1249  3860
1  [0.8873972  0.89024228 0.88868637 0.89246499 0...  21820  1190  1805  3304
2  [0.93696377 0.93696377 0.93669704 0.93874194 0...  22272   738   951  4158
3  [0.93660814 0.93807513 0.93705268 0.93994221 0...  22206   804   882  4227
4  [0.93798622 0.93834185 0.93847522 0.94043121 0...  22255   755   910  4199
5  [0.93474105 0.9331407  0.93491887 0.93536341 0...  22336   674  1098  4011
```

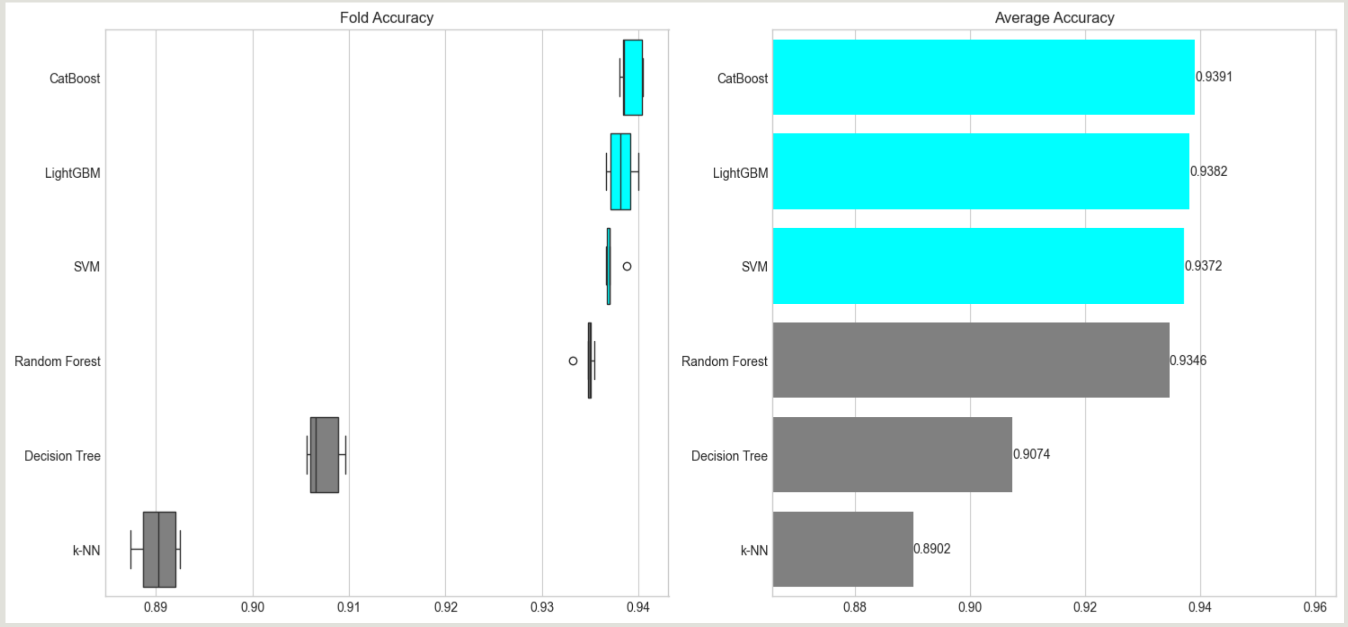
Having in account the following image we can make some conclusions, namely:

- **CatBoost** and **LightGBM** are the top-performing models.
- **SVM** is the most computationally expensive, with a **training time of 405.89s** and **testing time of 103.64s**, which is not practical for large-scale or real-time scenarios.
- **k-NN** has extremely fast **training time** (0.1086s) but a much **slower testing time** (6.5189s) due to its instance-based nature.
- To this dataset, **k-NN** in its current state is **not a valid option**, since it shows variability and lower CV accuracy scores.

Evaluation and comparison of models


Training Time by Model (seconds)
Training Time (s)
Decision Tree
k-NN
SVM
LightGBM
CatBoost
Random Forest


Prediction Time by Model (seconds)
Prediction Time (s)
Decision Tree
k-NN
SVM
LightGBM
CatBoost
Random Forest


Fold Accuracy
CatBoost
LightGBM
SVM
Random Forest
Decision Tree
k-NN
Average Accuracy
CatBoost
LightGBM
SVM
Random Forest
Decision Tree
k-NN