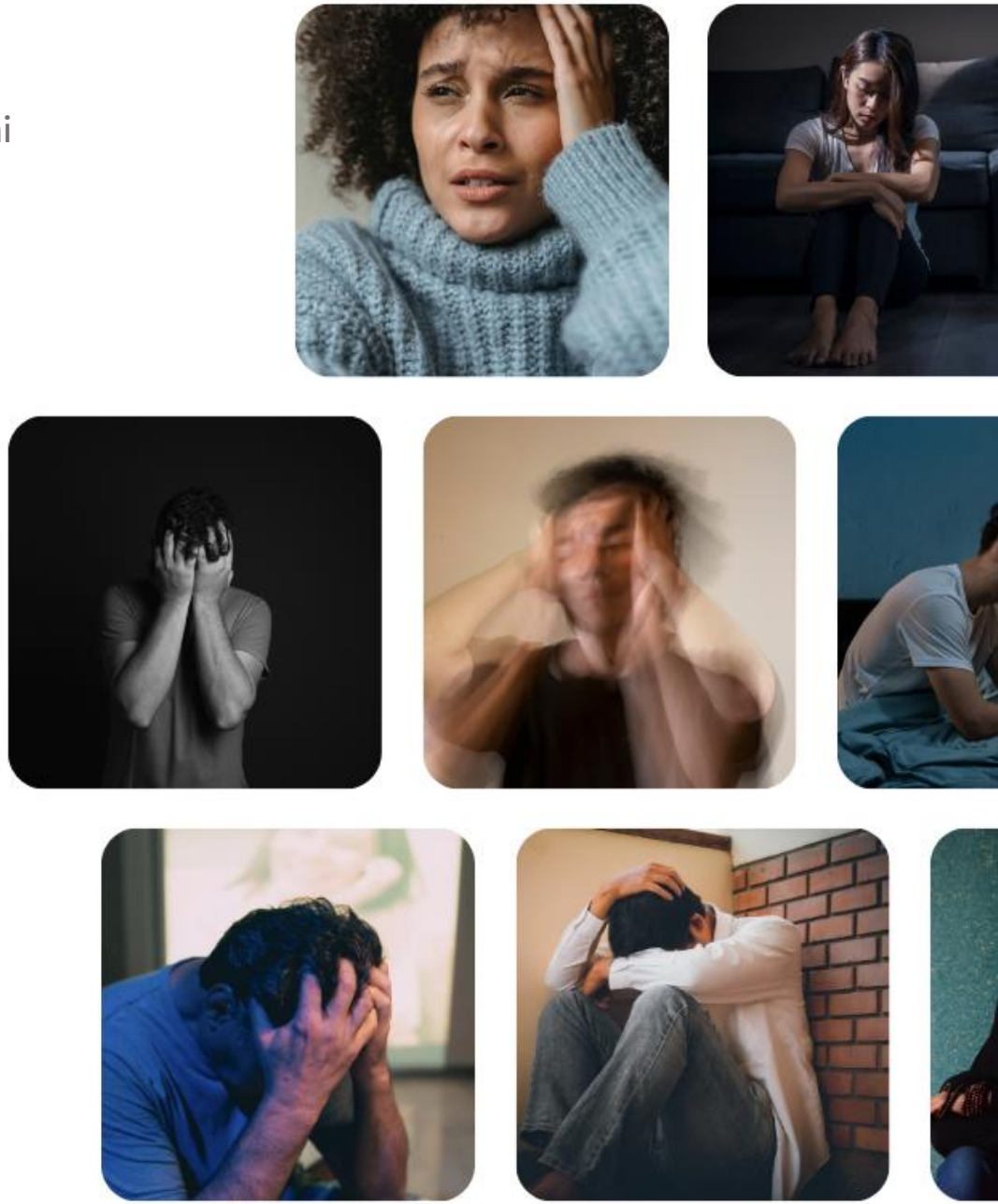
PGPDSE – Hyd, Mar'24
TeamONE
Dileep, Praneeth, Sreekar, Ghouse, Sreehitha, Kalyani
Mentored by - Ankush Bansal



Leveraging Machine Learning for Depression Screening

A Step Towards Proactive Intervention

Dataset

Converting from *.csv to *.parquet

Parquet is an open-source columnar storage file format designed for efficient data processing. Unlike CSV, which stores data row-by-row, Parquet stores data column-by-column. This columnar organization offers several key advantages

- Faster data loading
- Reduced storage space
- Improved data integrity
- Enhanced performance for machine learning tasks

<.py code>

```python
import pyarrow as pa
import pyarrow.parquet as pq

file = pd.read_csv("depression_data.csv")
table = pa.Table.from_pandas(file)
pq.write_table(table, "depression_data.parquet")

csv_df = pd.read_csv("depression_data.csv")
# to read csv file, python took - 0.0 Mins 0.746 Sec

parquet_df = pd.read_parquet("depression_data.parquet")
# to read parquet file, python took - 0.0 Mins 0.457 Sec

df = parquet_df
```

*.parquet file took **40%** less time to load that *.csv file

Dataset

	Name	Age	Marital Status	Education Level	Number of Children	Smoking Status	Physical Activity Level	Employment Status	Income	Alcohol Consumption	Dietary Habits	Sleep Patterns	History of Mental Illness	History of Substance Abuse	Family History of Depression	Depression
0	Christine Barker	31	Married	Bachelor's Degree	2	Non-smoker	Active	Unemployed	26265.67	Moderate	Moderate	Fair	Yes	No	Yes	Yes
1	Jacqueline Lewis	55	Married	High School	1	Non-smoker	Sedentary	Employed	42710.36	High	Unhealthy	Fair	Yes	No	No	Yes
2	Shannon Church	78	Widowed	Master's Degree	1	Non-smoker	Sedentary	Employed	125332.79	Low	Unhealthy	Good	No	No	Yes	No

Shape

- Rows : 4,13,768
- Columns : 16

Dtypes Columns

- Categorical : 13
- Numerical : 2
- Object : 1

Variable Name	Type	Description	Values
Age	Numerical	The age of the individual.	Numbers
Marital Status	Categorical	The marital status of the individual.	Married, Single, Widowed, Divorced
Education Level	Categorical	The highest level of education attained by the individual.	Bachelor's Degree, High School, Associate Degree, Master's Degree, PhD
Number of Children	Numerical	The number of children the individual has.	0, 1, 2, 3, 4
Smoking Status	Categorical	Whether the individual smokes.	Non-smoker, Former, Current
Physical Activity Level	Categorical	The level of physical activity engaged in by the individual.	Sedentary, Moderate, Active
Employment Status	Categorical	The current employment status of the individual.	Employed, Unemployed
Income	Numerical	The annual income of the individual.	Numbers
Alcohol Consumption	Categorical	The level of alcohol consumption by the individual.	Moderate, Low, High
Dietary Habits	Categorical	The eating habits of the individual.	Unhealthy, Moderate, Healthy
Sleep Patterns	Categorical	The sleep patterns of the individual.	Fair, Poor, Good
History of Mental Illness	Categorical	Whether the individual has a history of mental illness.	No, Yes
History of Substance Abuse	Categorical	Whether the individual has a history of substance abuse.	No, Yes
Family History of Depression	Categorical	Whether there is a family history of depression.	No, Yes
Depression	Categorical	Whether the individual has any chronic medical conditions.	No, Yes
Name	Object	Name of Individual	—

Nulls & Duplicates

Why..?

- Algorithm Compatibility
- Biased Analysis
- Reduced Statistical Power

➢ There are no Null Values in this Dataset

➢ There are no Duplicate Values in this Dataset

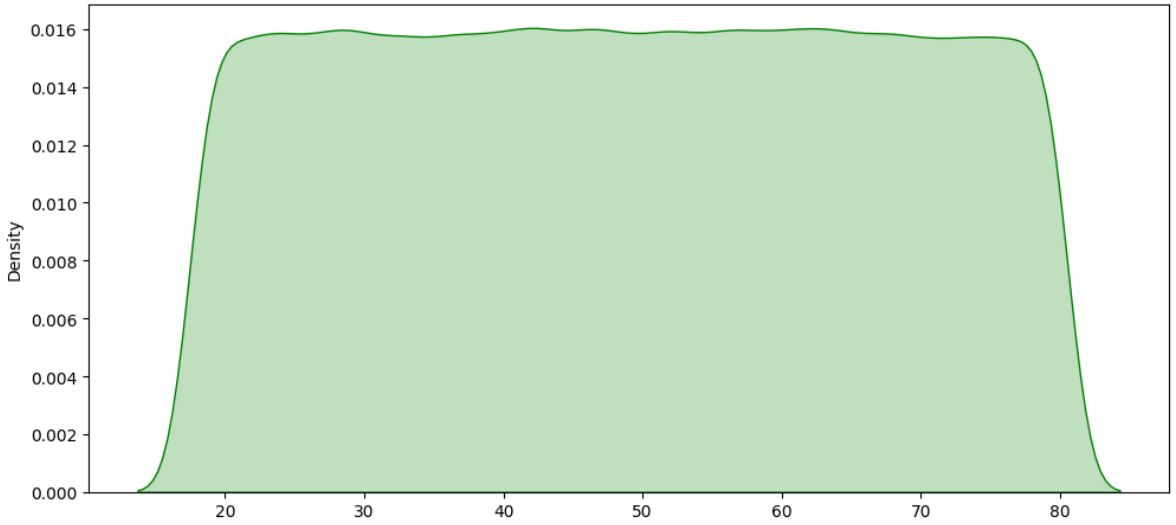
<.py code>

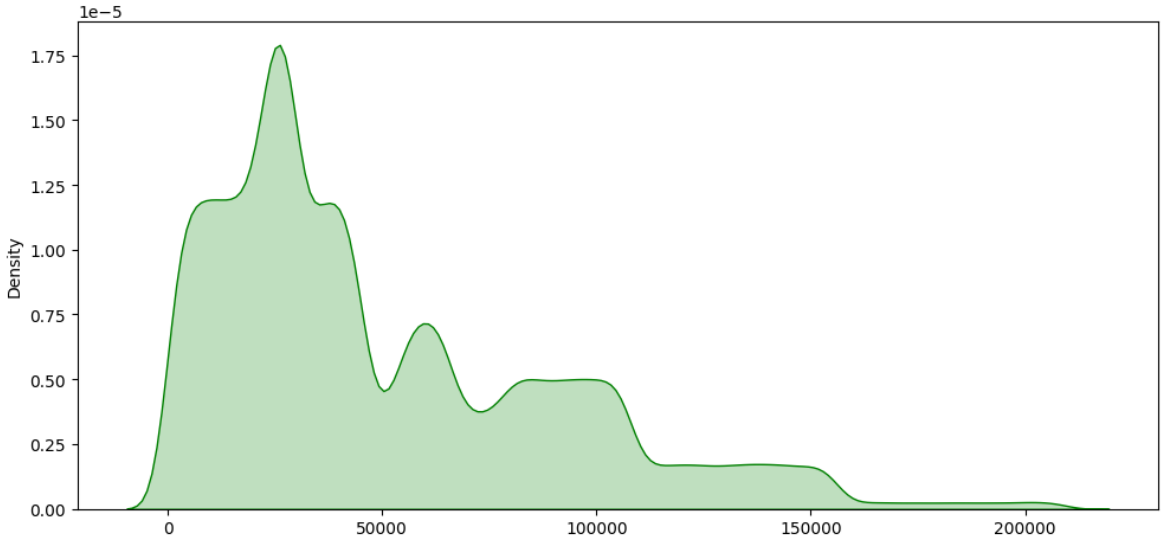
```
df.duplicated().sum() #Output - 0

df.isnull().sum()
```

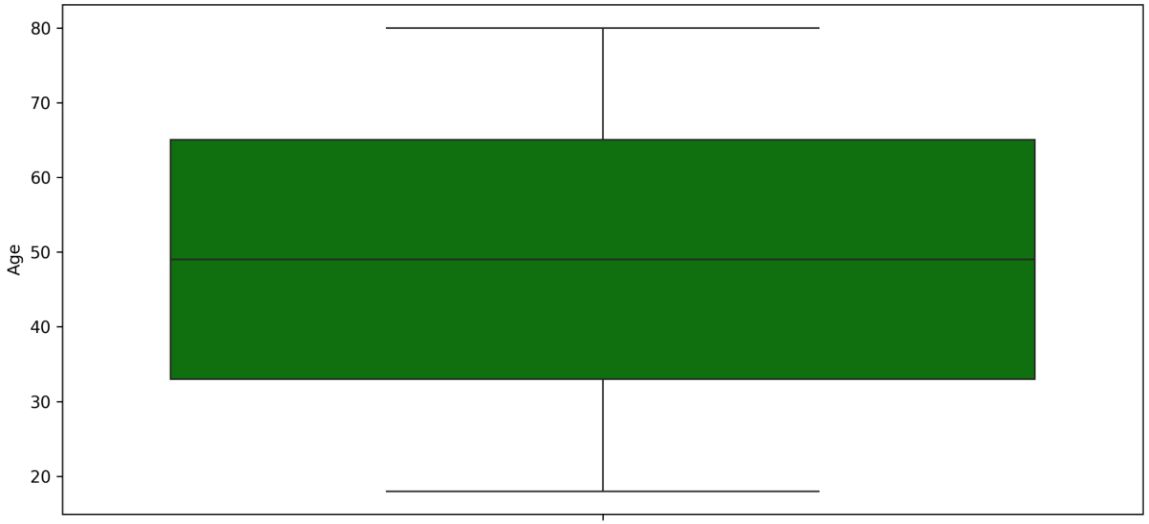
```
Name                            0
Age                             0
Marital Status                  0
Education Level                 0
Number of Children              0
Smoking Status                  0
Physical Activity Level         0
Employment Status               0
Income                          0
Alcohol Consumption             0
Dietary Habits                  0
Sleep Patterns                  0
History of Mental Illness       0
History of Substance Abuse      0
Family History of Depression    0
Depression                      0
```

Univariant Analysis

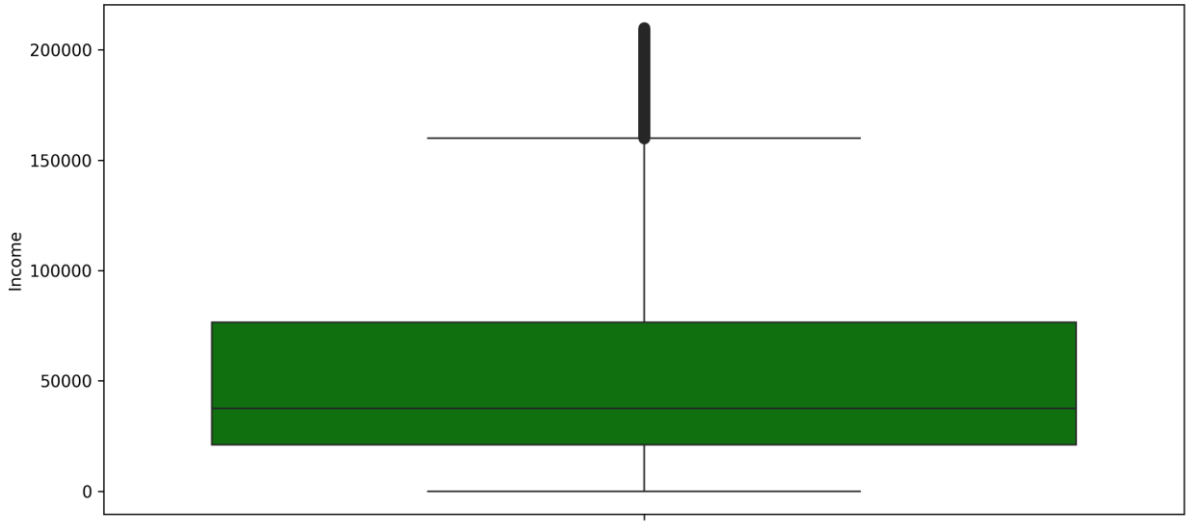




AGE



INCOME



Univariant Analysis

Marital Status

Category	Percentage
Married	58.11%
Single	17.43%
Widowed	16.55%
Divorced	7.91%

Education Level

Category	Percentage
Bachelor's Degree	30.05%
High School	28.74%
Associate Degree	19.33%
Master's Degree	17.83%
PhD	4.05%

Number of Children

Category	Percentage
0	37.52%
2	20.29%
1	20.28%
3	18.60%
4	3.31%

Smoking Status

Category	Percentage
Non-smoker	59.80%
Former	28.08%
Current	12.12%

Physical Activity Level

Category	Percentage
Sedentary	42.74%
Moderate	38.19%
Active	19.07%

Employment Status

Category	Percentage
Employed	64.20%
Unemployed	35.80%

Alcohol Consumption

Category	Percentage
Moderate	41.92%
Low	33.65%
High	24.43%

Dietary Habits

Category	Percentage
Unhealthy	41.28%
Moderate	41.19%
Healthy	17.52%

Sleep Patterns

Category	Percentage
Fair	47.56%
Poor	31.32%
Good	21.12%

History of Mental Illness

Category	Percentage
No	69.59%
Yes	30.41%

History of Substance Abuse

Category	Percentage
No	68.85%
Yes	31.15%

Family History of Depression

Category	Percentage
No	73.11%
Yes	26.89%

Depression

Category	Percentage
No	67.08%
Yes	32.92%

Scaling

Why..?

- Equalizing Feature Influence
- Improving Algorithm Performance
- Preventing Domination by Large Values:

<.py code>

```
scaler_objects = {}

for i in x_train.select_dtypes(include="number").columns:
    scaler_objects[i] = MinMaxScaler()
    scaler_objects[i].fit(x_train[[i]])
    x_train[i] = scaler_objects[i].transform(x_train[[i]])
```

We used “MinMax Scaler”, because of the outliers in “income” variable.

We have done “fit_transform” on training data, and just “transform” on testing data to prevent data leakage. Because the test data should be hidden from the model so that overfitting will not happen.

Before Scaling

Age	Income
33	34048.17
47	9002.91
77	121705.51
19	34575.37
61	12004.30

After Scaling

Age	Income
0.241935	0.162144
0.467742	0.042872
0.951613	0.579589
0.016129	0.164654
0.693548	0.057165

Encoding

Why..?

- Algorithm Compatibility

We used “One Hot Encoding” to make 14 categorical variables as numerical.

We have done “fit_transform” on training data, and just “transform” on testing data to prevent data leakage. Because the test data should be hidden from the model so that overfitting will not happen.

<.py code>

```
new_df = x_train.select_dtypes(include="number")
new_df.reset_index(inplace=True)
encoder_objects = {}

for i in x_train.select_dtypes(exclude="number").columns:
    # print(i)
    encoder_objects[i] = OneHotEncoder(dtype='int', drop="first")
    dummy_df =
encoder_objects[i].fit_transform(x_train[[i]]).toarray()
    new_df = pd.concat([new_df, pd.DataFrame(dummy_df,
columns=encoder_objects[i].get_feature_names_out())], axis=1)
    # print(new_df)

x_train = new_df
```

Base Model

We have use “Logistic Regression” is a base model. It gave 67% accuracy, but the model failed to detect class 1 datapoints in test data.

Possible Reasons for failing of algorithm:

1. Class Imbalance
2. Insufficient Features
3. Model Limitations

Category	Percentage
No	67.08%
Yes	32.92%

<.py code>

```
model = LogisticRegression()
model.fit(x_train, y_train)

y_pred = model.predict(x_test)
print(classification_report(y_test, y_pred))
```

Classification Report

	precision	recall	f1-score	support
0	0.67	1.00	0.80	55526
1	0.00	0.00	0.00	27228
accuracy			0.67	82754
macro avg	0.34	0.50	0.40	82754
weighted avg	0.45	0.67	0.54	82754

Works to be done:

1. Will check if this data is satisfying the assumption of Linear Regression?
    1. Linearity
    2. Independence of Residuals
    3. Homoscedasticity
    4. Normality of Residuals
    5. No Multicollinearity
2. Will do statistical tests to know which feature is important to predict target
3. Will do PCA to make sure there is no Multicollinearity.
4. Will do transformation on Numerical features
5. Will do hyperparameter tunning
6. Will check is this data performs better in clustering rather that classification
7. Will try various other classification algorithms
8. Will to feature engineering
9. Will make this whole algorithm dynamic using OOPs – (Pipeline)

THANK YOU

Please ask your question

PGPDSE – Hyd, Mar'24

TeamONE

Dileep, Praneeth, Sreekar, Ghouse, Sreehitha, Kalyani

Mentored by - Ankush Bansal