Identifying Depression in Adults

by Brian Nguyen
at Metis

DATA AND TARGET VARIABLE

Data: National Health and Nutritional Examination Survey

Ten years of data (2009-2018)

Target: Feeling Down, Depressed, or Hopeless

Self-assessment of above emotions ***nearly every day***

OVERVIEW OF COLLECTED DATA

23,922 — Adults Examined (Age 20+)

828 — (+) Down, Depressed, Hopeless

16 — Features (Binary and Continuous)

DIVERSITY OF FEATURES

8 of 16 Shown Based on Importance per Category

DEMOGRAPHIC

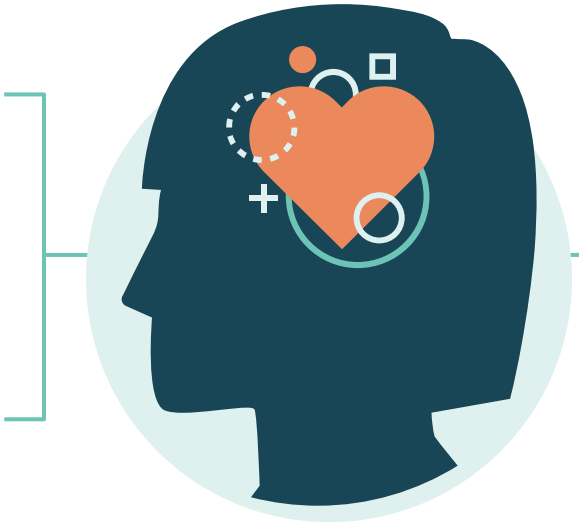
Normalized Income

Education Level

MENTAL

General Interest

Appetite Problems



PHYSICAL

Exercise

Body Mass Index (BMI)

OTHER

Smoking

Hours Sleeping

CURSORY MODEL COMPARISON

	Logistic Regression	Random Forest	Gaussian Naïve Bayes	XGBoost Classifier
Accuracy	0.968	0.969	0.916	0.967
Precision	0.596	0.631	0.214	0.545
Recall	0.260	0.225	0.527	0.264
F1 Score	0.359	0.330	0.303	0.355

Cross-Validation using K-Fold = 5

CURSORY MODEL COMPARISON

	Logistic Regression	Random Forest	Gaussian Naïve Bayes	XGBoost Classifier
Accuracy	0.968	0.969	0.916	0.967
Precision	0.596	0.631	0.214	0.545
Recall	0.260	0.225	0.527	0.264
F1 Score	0.359	0.330	0.303	0.355

Cross-Validation using K-Fold = 5

CURSORY MODEL COMPARISON

	Logistic Regression	Random Forest	Gaussian Naïve Bayes	XGBoost Classifier
Accuracy	0.968	0.969	0.916	0.967
Precision	0.596	0.631	0.214	0.545
Recall	0.260	0.225	0.527	0.264
F1 Score	0.359	0.330	0.303	0.355

Cross-Validation using K-Fold = 5

CURSORY MODEL COMPARISON

	Logistic Regression	Random Forest	Gaussian Naïve Bayes	XGBoost Classifier
Accuracy	0.968	0.969	0.916	0.967
Precision	0.596	0.631	0.214	0.545
Recall	0.260	0.225	0.527	0.264
F1 Score	0.359	0.330	0.303	0.355

Cross-Validation using K-Fold = 5

DETAILED COMPARISON AND RESULTS

Gaussian Naïve Bayes	Predicted (–)	Predicted (+)
Actual (–)	4222	397
Actual (+)	28	138

Recall = 83%

Oversampled

6x of (+) Class

DETAILED COMPARISON AND RESULTS

Gaussian Naïve Bayes	Predicted (–)	Predicted (+)
Actual (–)	4222	397
Actual (+)	28	138

Recall = 83%

Oversampled

6x of (+) Class

XGBoost Classifier$_{Tuned}$	Predicted (–)	Predicted (+)
Actual (–)	4237	382
Actual (+)	33	133

Recall = 80%

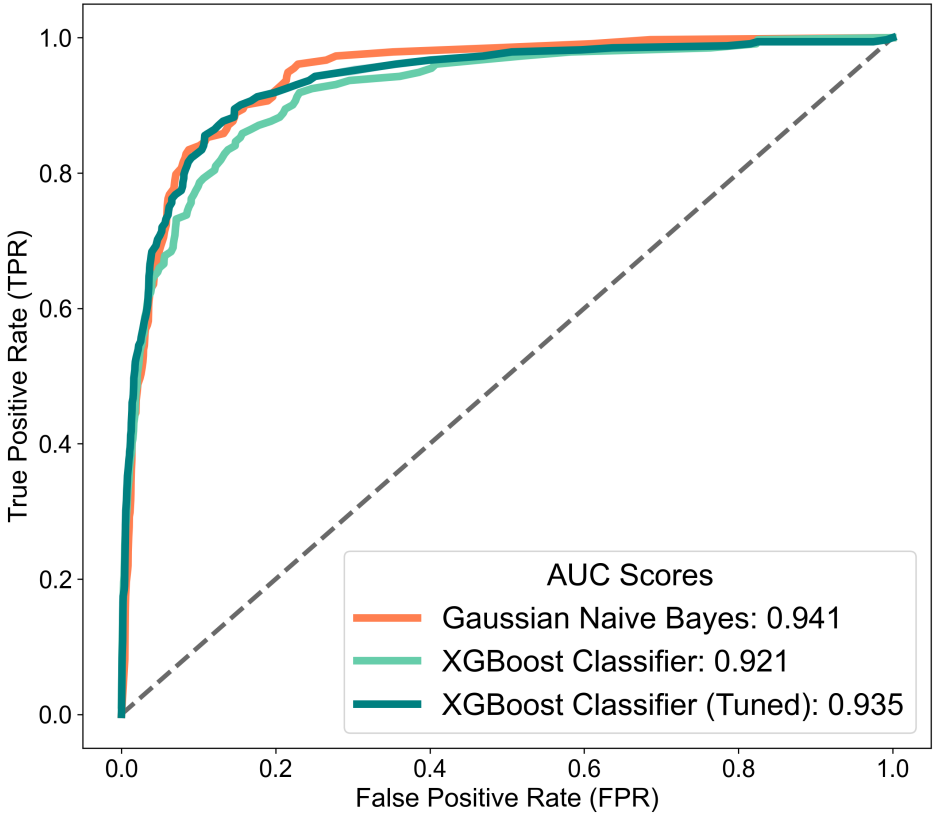
Not Oversampled

Hyperparameter Tuning

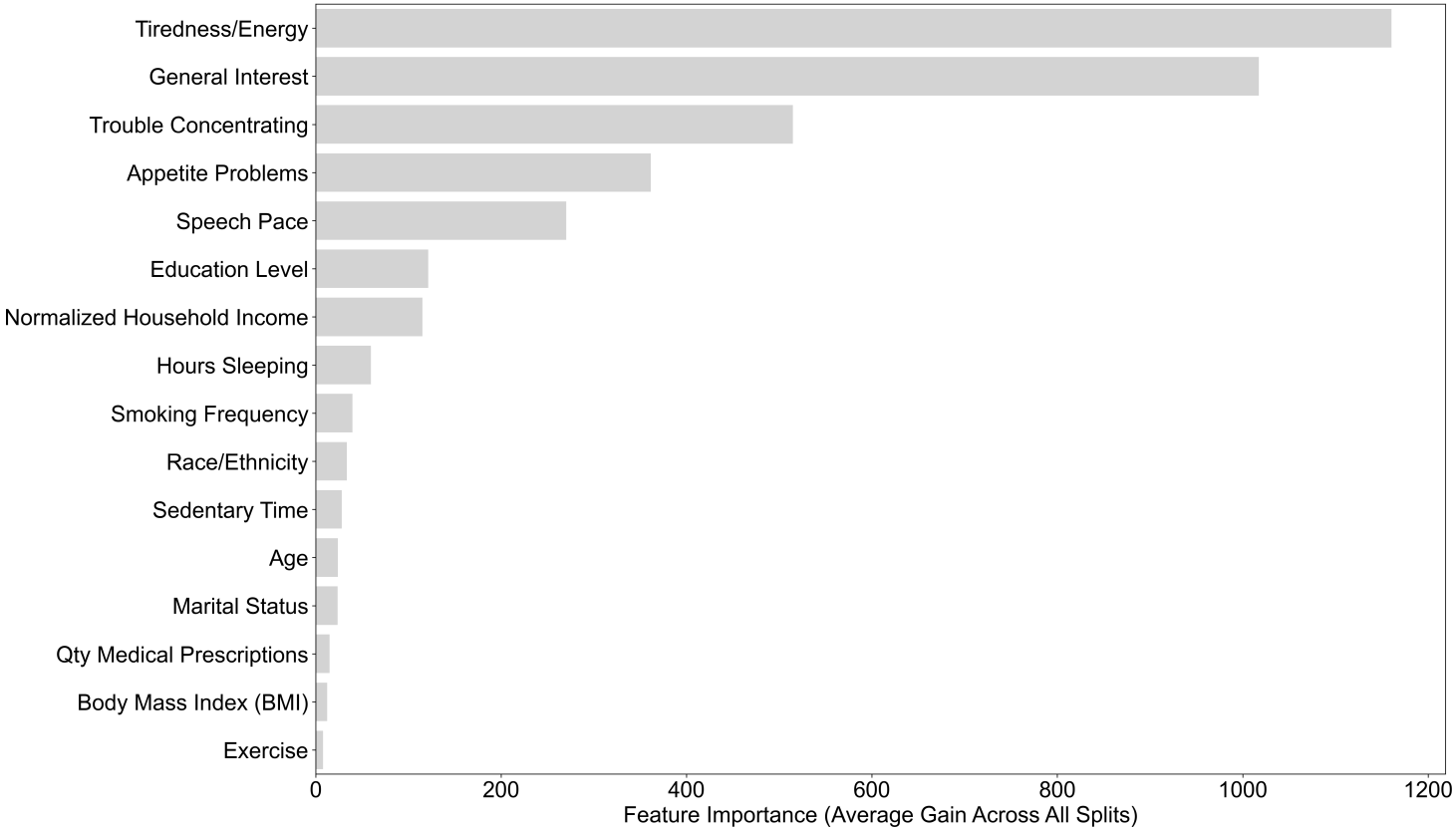
DETAILED COMPARISON AND RESULTS

Gaussian Naïve Bayes	Predicted (–)	Predicted (+)
Actual (–)	4222	397
Actual (+)	28	138

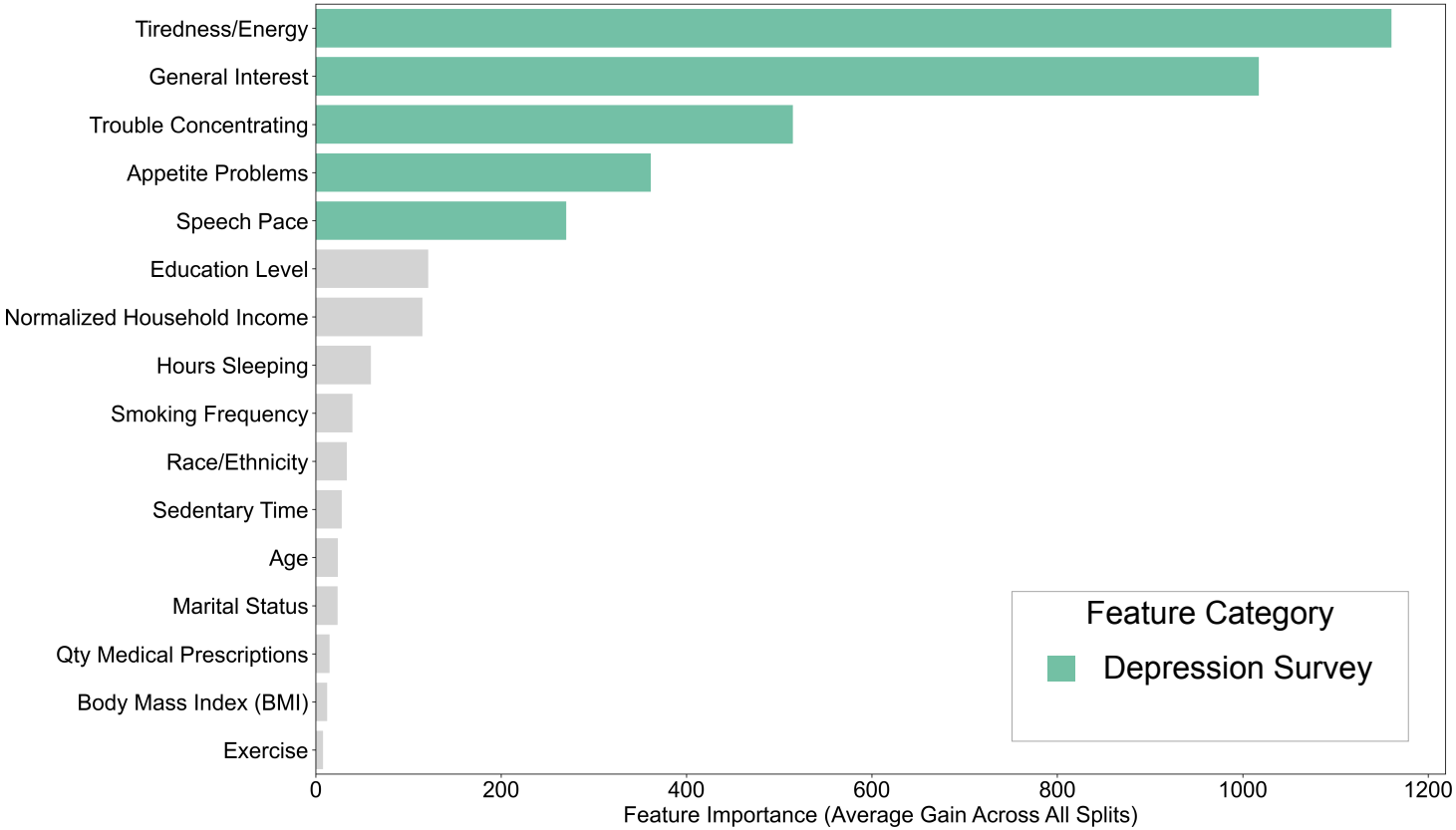
XGBoost Classifier$_{Tuned}$	Predicted (–)	Predicted (+)
Actual (–)	4237	382
Actual (+)	33	133



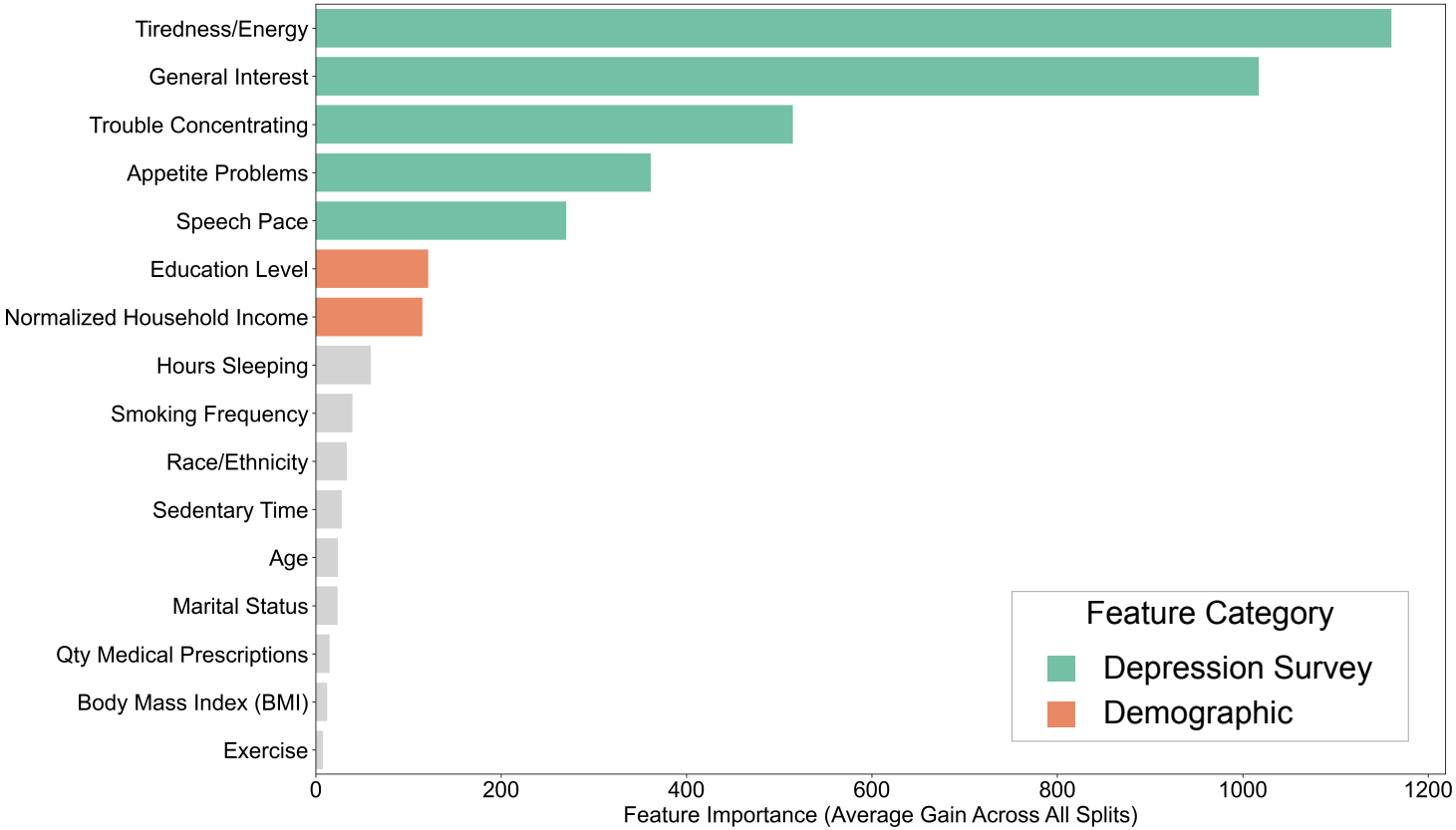
FEATURE IMPORTANCE



FEATURE IMPORTANCE



FEATURE IMPORTANCE



MONITORING DEPRESSION

MENTAL STATE

Aligned with scientific literature[1]

Matches clinical survey: *e.g.* tiredness and interest

DEMOGRAPHIC

Household income

Level of education

AGE

Poor accuracy for age 19 and under

Feature importance changes for age groups

[1]Kroenke K, Spitzer RL, William JB. The PHQ-9: validity of a brief depression severity measure. J Gen Intern Med 2001; 16: 1606-13.

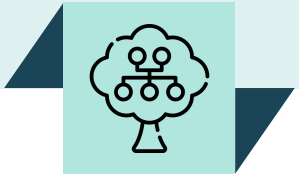
FUTURE WORK

AGE (19–)

Separate model

Explore features

e.g. trauma and family history



LIFESTYLE

Solely lifestyle features

Actionable

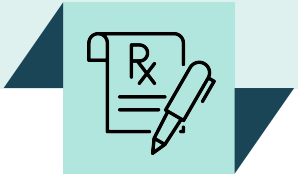
e.g. diet, exercise, and smoking

MULTICLASS

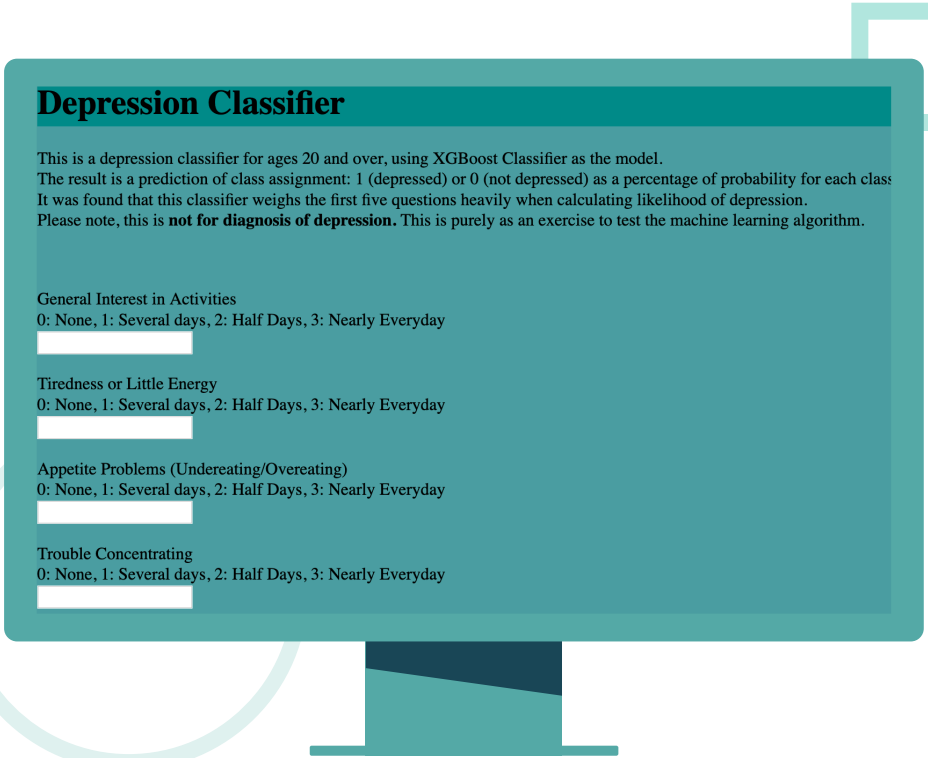
Anxiety and psychosis

Classify by prescriptions

e.g. lab and exam data



WEBSITE



https://depression-classifier-adults.herokuapp.com/

Thank You

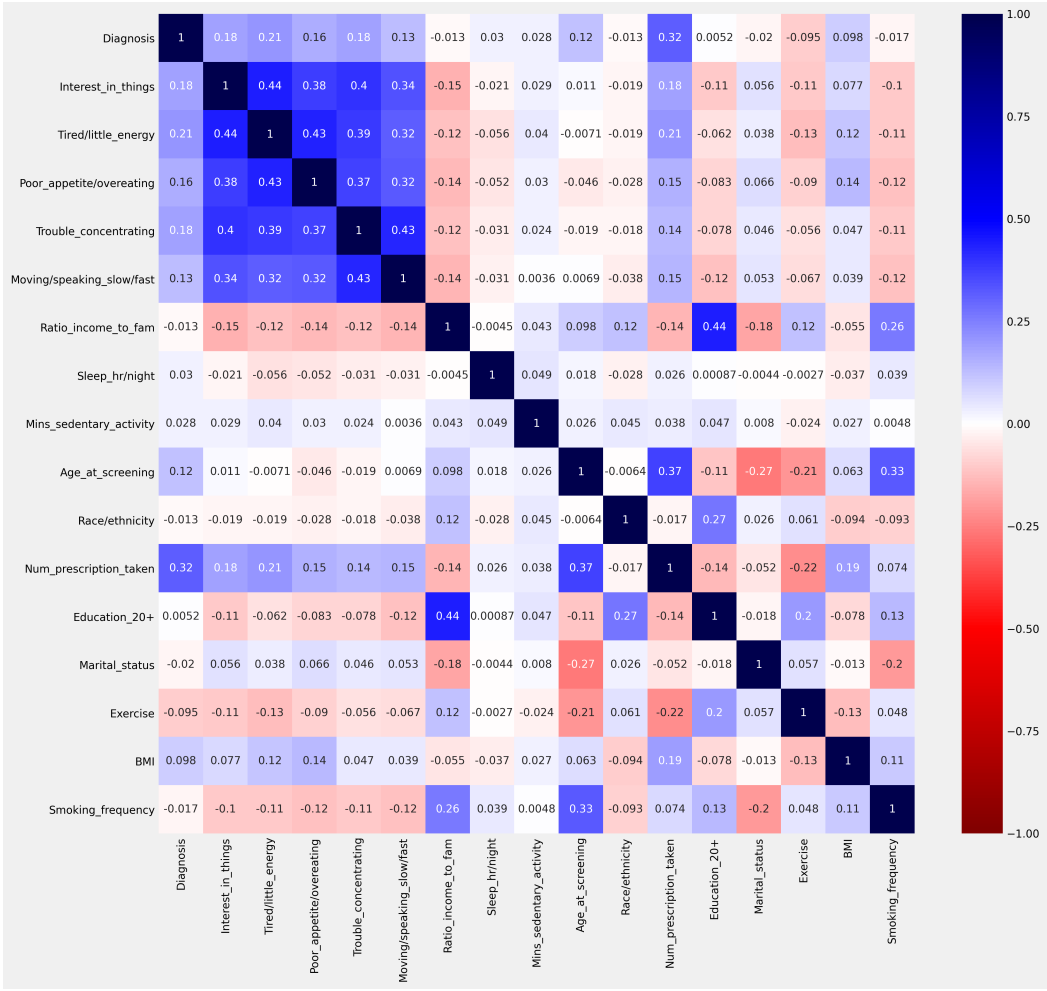
Please let me know if you have any questions.

Brian Nguyen
at Metis

github.com/bt-nguyen

APPENDIX

TABLE OF CONTENTS


Diagnosis 1 0.18 0.21 0.16 0.18 0.13 -0.013 0.03 0.028 0.12 -0.013 0.32 0.0052 -0.02 -0.095 0.098 -0.017
Interest_in_things 0.18 1 0.44 0.38 0.4 0.34 -0.15 -0.021 0.029 0.011 -0.019 0.18 -0.11 0.056 -0.11 0.077 -0.1
Tired/little_energy 0.21 0.44 1 0.43 0.39 0.32 -0.12 -0.056 0.04 -0.0071 -0.019 0.21 -0.062 0.038 -0.13 0.12 -0.11
Poor_appetite/overeating 0.16 0.38 0.43 1 0.37 0.32 -0.14 -0.052 0.03 -0.046 -0.028 0.15 -0.083 0.066 -0.09 0.14 -0.12
Trouble_concentrating 0.18 0.4 0.39 0.37 1 0.43 -0.12 -0.031 0.024 -0.019 -0.018 0.14 -0.078 0.046 -0.056 0.047 -0.11
Moving/speaking_slow/fast 0.13 0.34 0.32 0.32 0.43 1 -0.14 -0.031 0.0036 0.0069 -0.038 0.15 -0.12 0.053 -0.067 0.039 -0.12
Ratio_income_to_fam -0.013 -0.15 -0.12 -0.14 -0.12 -0.14 1 -0.0045 0.043 0.098 0.12 -0.14 0.44 -0.18 0.12 -0.055 0.26
Sleep_hr/night 0.03 -0.021 -0.056 -0.052 -0.031 -0.031 -0.0045 1 0.049 0.018 -0.028 0.026 0.00087 -0.0044 -0.0027 -0.037 0.039
Mins_sedentary_activity 0.028 0.029 0.04 0.03 0.024 0.0036 0.043 0.049 1 0.026 0.045 0.038 0.047 0.008 -0.024 0.027 0.0048
Age_at_screening 0.12 0.011 -0.0071 -0.046 -0.019 0.0069 0.098 0.018 0.026 1 -0.0064 0.37 -0.11 -0.27 -0.21 0.063 0.33
Race/ethnicity -0.013 -0.019 -0.019 -0.028 -0.018 -0.038 0.12 -0.028 0.045 -0.0064 1 -0.017 0.27 0.026 0.061 -0.094 -0.093
Num_prescription_taken 0.32 0.18 0.21 0.15 0.14 0.15 -0.14 0.026 0.038 0.37 -0.017 1 -0.14 -0.052 -0.22 0.19 0.074
Education_20+ 0.0052 -0.11 -0.062 -0.083 -0.078 -0.12 0.44 0.00087 0.047 -0.11 0.27 -0.14 1 -0.018 0.2 -0.078 0.13
Marital_status -0.02 0.056 0.038 0.066 0.046 0.053 -0.18 -0.0044 0.008 -0.27 0.026 -0.052 -0.018 1 0.057 -0.013 -0.2
Exercise -0.095 -0.11 -0.13 -0.09 -0.056 -0.067 0.12 -0.0027 -0.024 -0.21 0.061 -0.22 0.2 0.057 1 -0.13 0.048
BMI 0.098 0.077 0.12 0.14 0.047 0.039 -0.055 -0.037 0.027 0.063 -0.094 0.19 -0.078 -0.013 -0.13 1 0.11
Smoking_frequency -0.017 -0.1 -0.11 -0.12 -0.11 -0.12 0.26 0.039 0.0048 0.33 -0.093 0.074 0.13 -0.2 0.048 0.11 1
Diagnosis
Interest_in_things
Tired/little_energy
Poor_appetite/overeating
Trouble_concentrating
Moving/speaking_slow/fast
Ratio_income_to_fam
Sleep_hr/night
Mins_sedentary_activity
Age_at_screening
Race/ethnicity
Num_prescription_taken
Education_20+
Marital_status
Exercise
BMI
Smoking_frequency
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Models to Test in Functions ¶

:

```
log_model = LogisticRegression(max_iter = 1000,
                               solver = 'liblinear')
dtr_model = DecisionTreeClassifier()
rfc_model = RandomForestClassifier()
gnb_model = GaussianNB()
bnb_model = BernoulliNB()
gbm_model = xgb.XGBClassifier(objective = "binary:logistic")
gbm_tuned_model = xgb.XGBClassifier(n_estimators = 2700,
                                    max_depth = 1,
                                    objective = "binary:logistic",
                                    learning_rate = 0.1,
                                    subsample = 1,
                                    min_child_weight = 1,
                                    colsample_bytree = 1,
                                    scale_pos_weight = 18)
```

XGradient Boost Classifier: Hyperparameter Tuning

In [32]:

```
'''
This is a setup for hyperparameter tuning.
It uses a random grid search to find the 'best' hyperparamaters to use for random forest classification.
'''

from sklearn.model_selection import RandomizedSearchCV

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 1000, stop = 11000, num = 101)]

# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(1, 5, num = 5)]

scale_pos_weight = [int(x) for x in np.linspace(1, 20, num = 20)]

colsample_bytree = [round(x, 1) for x in np.linspace(0.2, 2.0, num = 10)]

# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_depth': max_depth,
               'scale_pos_weight': scale_pos_weight,
               'colsample_bytree': colsample_bytree}

print(random_grid)
```

```
{'n_estimators': [1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000, 2100, 2200, 2300, 2400, 2500, 2600, 2700, 2800,
2900, 3000, 3100, 3200, 3300, 3400, 3500, 3600, 3700, 3800, 3900, 4000, 4100, 4200, 4300, 4400, 4500, 4600, 4700, 4800, 4900, 5000,
5100, 5200, 5300, 5400, 5500, 5600, 5700, 5800, 5900, 6000, 6100, 6200, 6300, 6400, 6500, 6600, 6700, 6800, 6900, 7000, 7100, 7200,
7300, 7400, 7500, 7600, 7700, 7800, 7900, 8000, 8100, 8200, 8300, 8400, 8500, 8600, 8700, 8800, 8900, 9000, 9100, 9200, 9300, 9400,
9500, 9600, 9700, 9800, 9900, 10000, 10100, 10200, 10300, 10400, 10500, 10600, 10700, 10800, 10900, 11000], 'max_depth': [1, 2, 3,
4, 5], 'scale_pos_weight': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20], 'colsample_bytree': [0.2, 0.4,
0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0]}
```

```
10 model_comparison(X_train, y_train)
```

```
Logistic Regression:
 [[4591   28]
 [ 118   48]]
Decision Tree Classifier:
 [[4486  133]
 [ 116   50]]
Random Forest Classifier:
 [[4604   15]
 [ 122   44]]
Bernoulli Naive Bayes:
 [[4269  350]
 [  63  103]]
Gaussian Naive Bayes:
 [[4294  325]
 [  34  132]]
Extreme Gradient Boost:
 [[4583   36]
 [ 113   53]]
Extreme Gradient Boost (Tuned):
 [[4237  382]
 [  33  133]]
```

21]:

	Logistic Regression	Decision Tree	Random Forest	Bernoulli Naive Bayes	Gaussian Naive Bayes	XGradient Boost Classifier	XGradient Boost Classifier (Tuned)	Best Score
Accuracy	0.969488	0.947962	0.971369	0.913689	0.924974	0.968861	0.913271	Random Forest
Precision	0.631579	0.273224	0.745763	0.227373	0.288840	0.595506	0.258252	Random Forest
Recall	0.289157	0.301205	0.265060	0.620482	0.795181	0.319277	0.801205	XGradient Boost Classifier (Tuned)
F1 Score	0.396694	0.286533	0.391111	0.332795	0.423756	0.415686	0.390602	Gaussian Naive Bayes

```
3 model_comparison(X_train_resample, y_train_resample)
```

```
Logistic Regression:
 [[4436  183]
 [  51  115]]
Decision Tree Classifier:
 [[4486  133]
 [ 104   62]]
Random Forest Classifier:
 [[4589   30]
 [ 112   54]]
Bernoulli Naive Bayes:
 [[4095  524]
 [  45  121]]
Gaussian Naive Bayes:
 [[4222  397]
 [  28  138]]
Extreme Gradient Boost:
 [[4552   67]
 [  99   67]]
Extreme Gradient Boost (Tuned):
 [[3583 1036]
 [  11  155]]
```

[22]:

	Logistic Regression	Decision Tree	Random Forest	Bernoulli Naive Bayes	Gaussian Naive Bayes	XGradient Boost Classifier	XGradient Boost Classifier (Tuned)	Best Score
Accuracy	0.951097	0.950470	0.970324	0.881087	0.911181	0.965308	0.781191	Random Forest
Precision	0.385906	0.317949	0.642857	0.187597	0.257944	0.500000	0.130143	Random Forest
Recall	0.692771	0.373494	0.325301	0.728916	0.831325	0.403614	0.933735	XGradient Boost Classifier (Tuned)
F1 Score	0.495690	0.343490	0.432000	0.298397	0.393723	0.446667	0.228445	Logistic Regression

```
3 model_comparison_crossval(X_train, y_train, 5)
```

3]:

	Logistic Regression	Decision Tree	Random Forest	Bernoulli Naive Bayes	Gaussian Naive Bayes	XGradient Boost Classifier	XGradient Boost Classifier (Tuned)	Best Score
Accuracy	0.968229	0.947588	0.968856	0.923395	0.915713	0.966661	0.915765	Random Forest
Precision	0.595992	0.280015	0.655391	0.272712	0.214001	0.544921	0.256330	Random Forest
Recall	0.259923	0.327785	0.225120	0.726635	0.527136	0.264377	0.753759	XGradient Boost Classifier (Tuned)
F1 Score	0.359387	0.301753	0.332112	0.396504	0.303447	0.354856	0.382527	Bernoulli Naive Bayes

In [12]:

```
# We will assign NaN values in the depression survey as 9: don't know
# This follows the line of thought that not filling it out most closely aligns with not knowing how to respond
# (Alternatively we can assign it 7, meaning refuse to answer, but that suggests a firm response)

df.loc[(pd.isna(df['Interest_in_things'])), 'Interest_in_things'] = 99
#df.loc[(pd.isna(df['Down_depressed_hopeless'])), 'Down_depressed_hopeless'] = 99
#df.loc[(pd.isna(df['Trouble/too_much_sleeping'])), 'Trouble/too_much_sleeping'] = 99
df.loc[(pd.isna(df['Tired/little_energy'])), 'Tired/little_energy'] = 99
df.loc[(pd.isna(df['Poor_appetite/overeating'])), 'Poor_appetite/overeating'] = 99
#df.loc[(pd.isna(df['Feeling_bad'])), 'Feeling_bad'] = 99
df.loc[(pd.isna(df['Trouble_concentrating'])), 'Trouble_concentrating'] = 99
df.loc[(pd.isna(df['Moving/speaking_slow/fast'])), 'Moving/speaking_slow/fast'] = 99
#df.loc[(pd.isna(df['Thoughts_of_death'])), 'Thoughts_of_death'] = 99
#df.loc[(pd.isna(df['Difficulty_from_problems'])), 'Difficulty_from_problems'] = 99


df.loc[(pd.isna(df['Smoking_frequency'])), 'Smoking_frequency'] = 99
df.loc[(pd.isna(df['Num_prescription_taken'])), 'Num_prescription_taken'] = 99
```

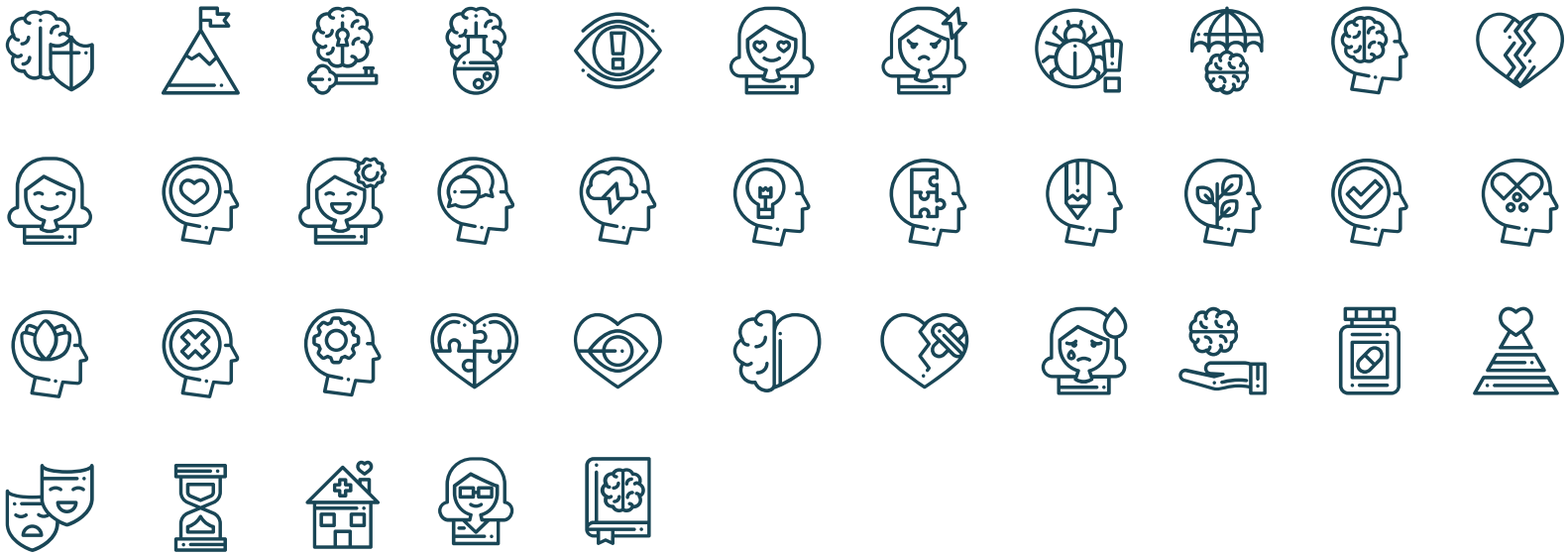
In [13]:

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

Out[13]:

```
Diagnosis                    0
Interest_in_things           0
Tired/little_energy          0
Poor_appetite/overeating     0
Trouble_concentrating        0
Moving/speaking_slow/fast    0
Ratio_income_to_fam       2052
Sleep_hr/night            9366
Mins_sedentary_activity     31
Age_at_screening             0
Race/ethnicity               0
Num_prescription_taken       0
Education_20+                0
Marital_status               0
Exercise                     0
BMI                          0
Smoking_frequency            0
dtype: int64
```

ALTERNATIVE ICONS



ALTERNATIVE RESOURCES

RESOURCES

Did you like the resources on this template? Get them for free at our other websites.

VECTORS

- Modern mental health concept with flat design
- Modern mental health concept with flat design 2

PHOTOS

- Patient on the divan
- Patient with headache

ICONS

- Psychology Icon Pack

CONTENTS OF THIS TEMPLATE

Here's what you'll find in this **Slidesgo** template:

1. A slide structure based on a thesis defense presentation, which you can easily adapt to your needs. For more info on how to edit the template, please visit **Slidesgo School** or read our **FAQs**.
2. An assortment of illustrations that are suitable for use in the presentation can be found in the **alternative resources** slide.
3. A **thanks slide**, which you must keep so that proper credits for our design are given.
4. A **resources slide**, where you'll find links to all the elements used in the template.
5. **Instructions for use**.
6. Final slides with:
    7. The **fonts and colors** used in the template.
    8. More **infographic resources**, whose size and color can be edited.
    9. Sets of **customizable icons** of the following themes: general, business, avatar, creative process, education, help & support, medical, nature, performing arts, SEO & marketing, and teamwork.

You can delete this slide when you're done editing the presentation.

Instructions for use

In order to use this template, you must credit Slidesgo by keeping the Thanks slide.

You are allowed to:

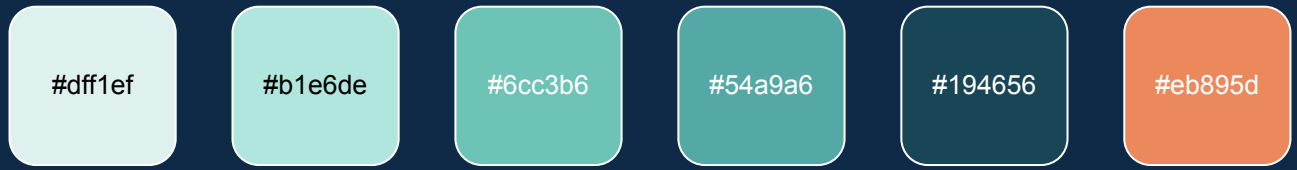
For more information about editing slides, please read our FAQs or visit Slidesgo School:
https://slidesgo.com/faqs and https://slidesgo.com/slidesgo-school

Fonts & colors used

This presentation has been made using the following fonts:

Raleway
(https://fonts.google.com/specimen/Raleway)

Roboto
(https://fonts.google.com/specimen/Roboto)

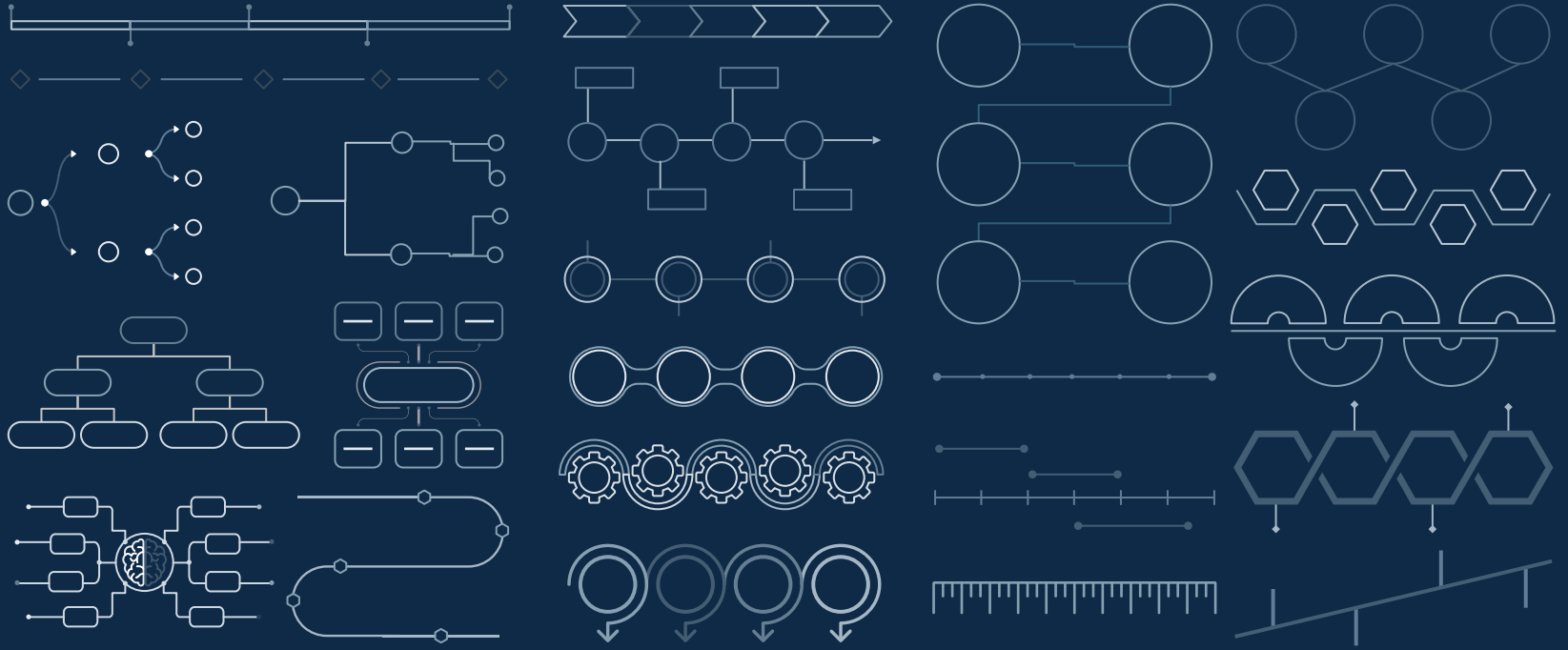


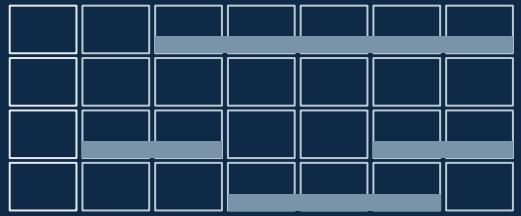
Use our editable graphic resources...

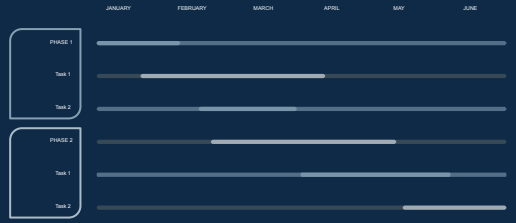
You can easily resize these resources without losing quality. To change the color, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Group the resource again when you're done.

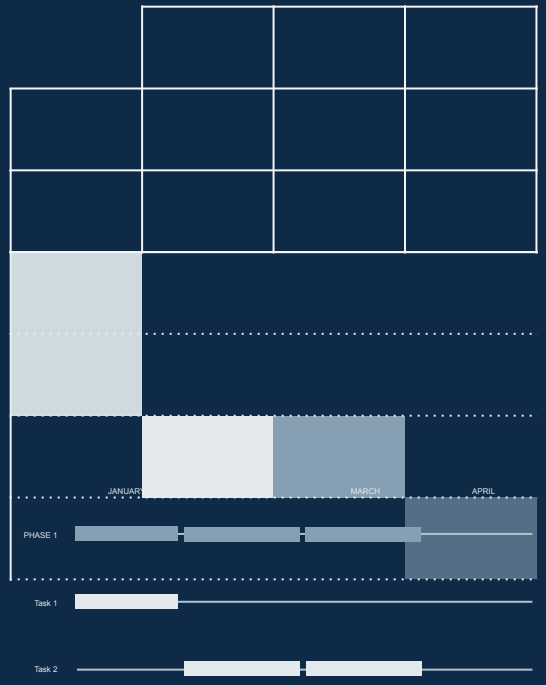


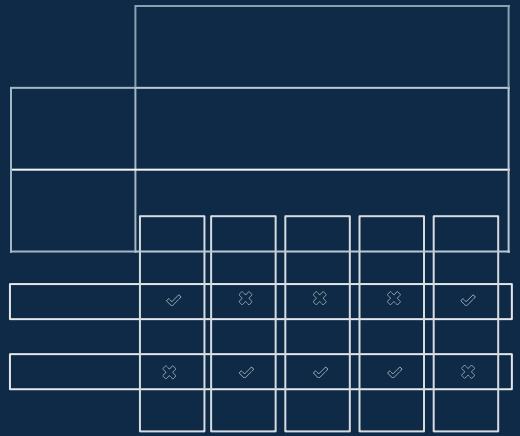


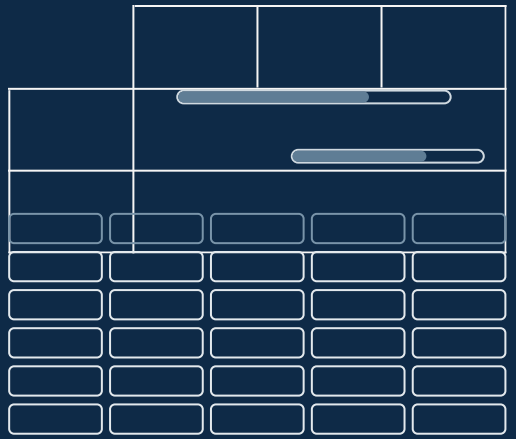



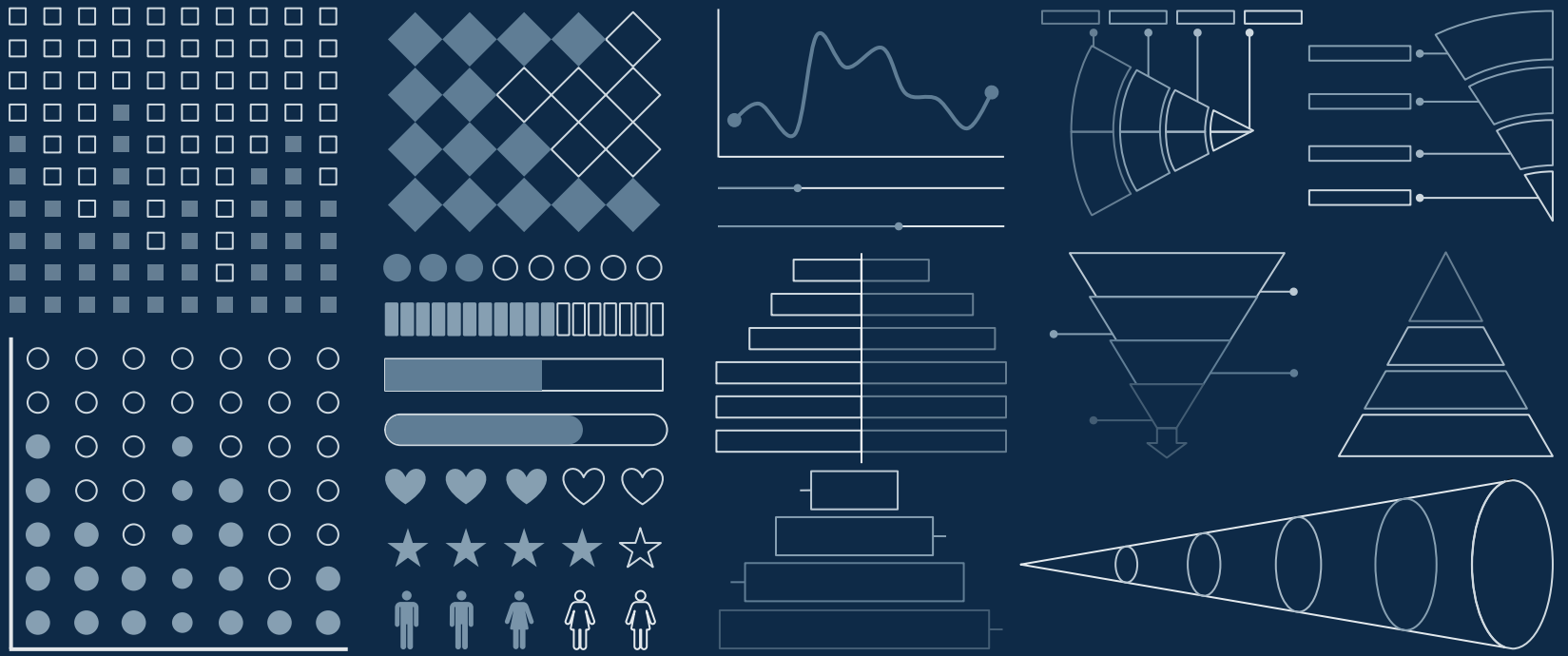



JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
PHASE 1
Task 1
Task 2
PHASE 2
Task 1
Task 2


JANUARY
MARCH
APRIL
PHASE 1
Task 1
Task 2







...and our sets of editable icons

You can resize these icons without losing quality.
You can change the stroke and fill color; just select the icon and click on the paint bucket/pen.
In Google Slides, you can also use Flaticon's extension, allowing you to customize and add even more icons.

Educational Icons

Medical Icons

Business Icons

Teamwork Icons

Help & Support Icons

Avatar Icons

Creative Process Icons

Performing Arts Icons

Nature Icons

SEO & Marketing Icons


slidesgo