Visiting Advertisements predictor

SHAI for AI

AGENDA

01

EXPLORATORY DATA ANALYSIS (EDA)

02

FEATURE ENGINEERING

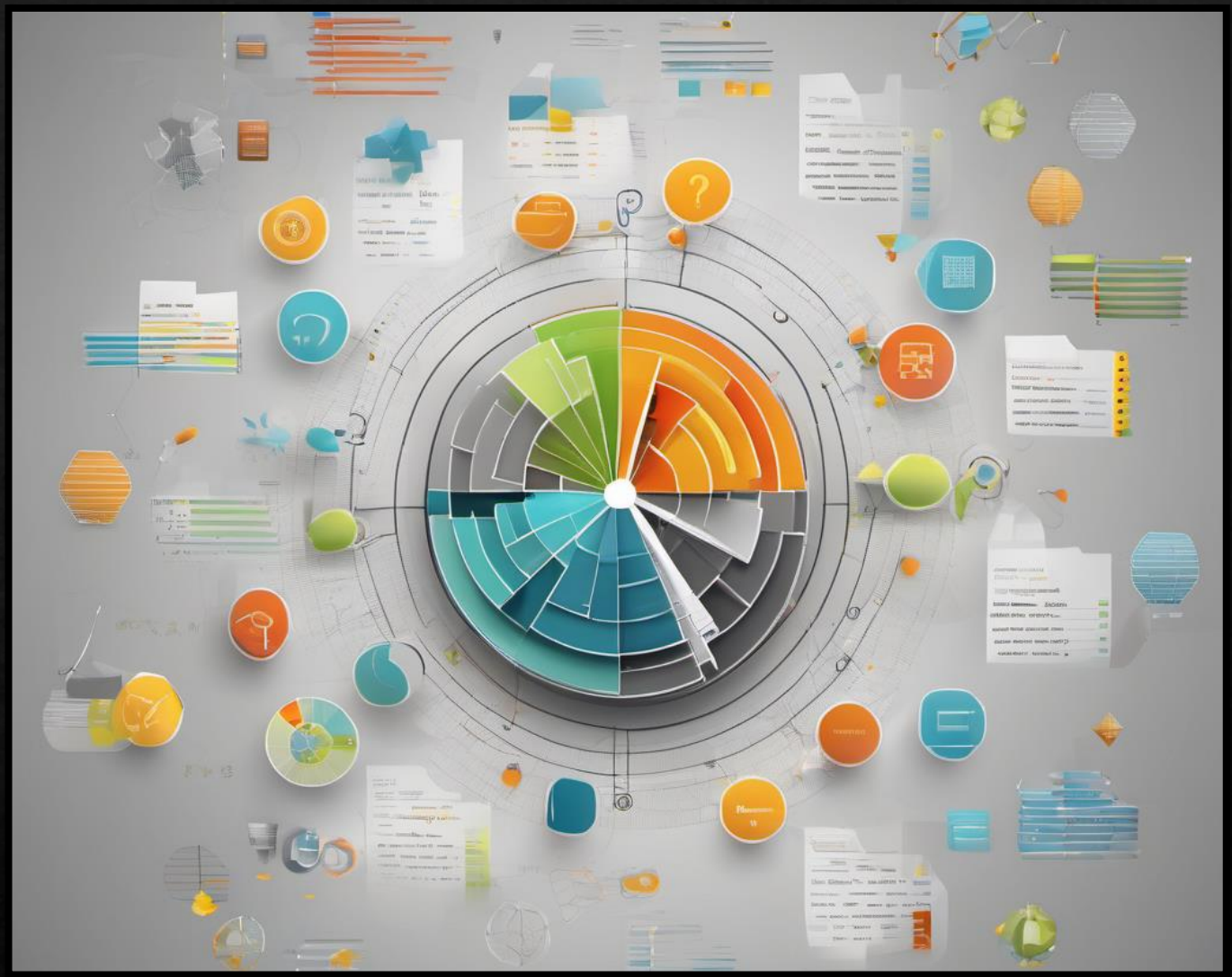
03

CHOOSING MODEL

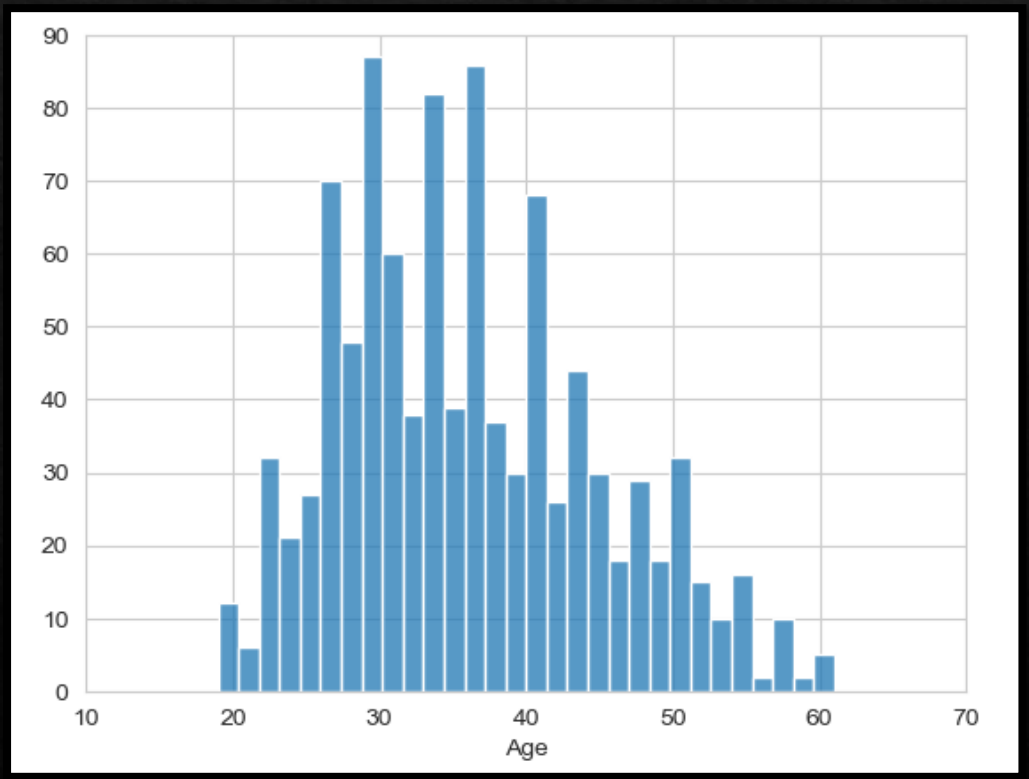
04

PREDICTION & EVALUATION

EXPLORATORY DATA ANALYSIS (EDA)

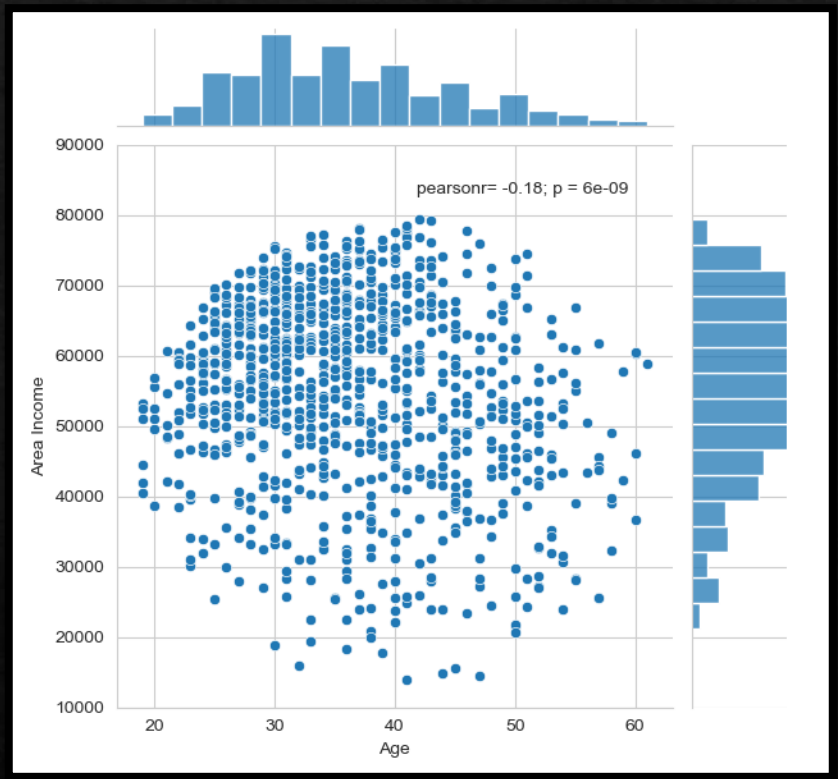


AGE HISTOGRAM



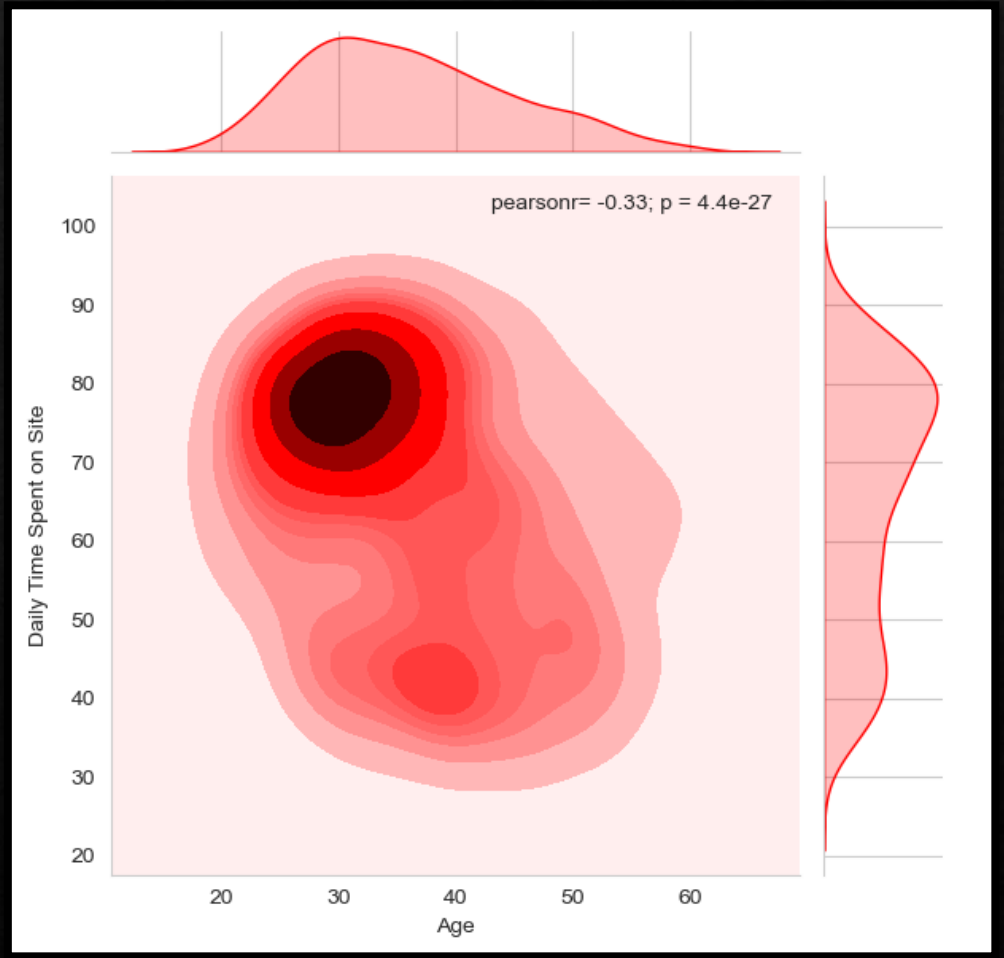
- We have noticed that most of the people who visit the website are between the ages of 30 and 40.

AGE – AREA INCOME JOINPLOT



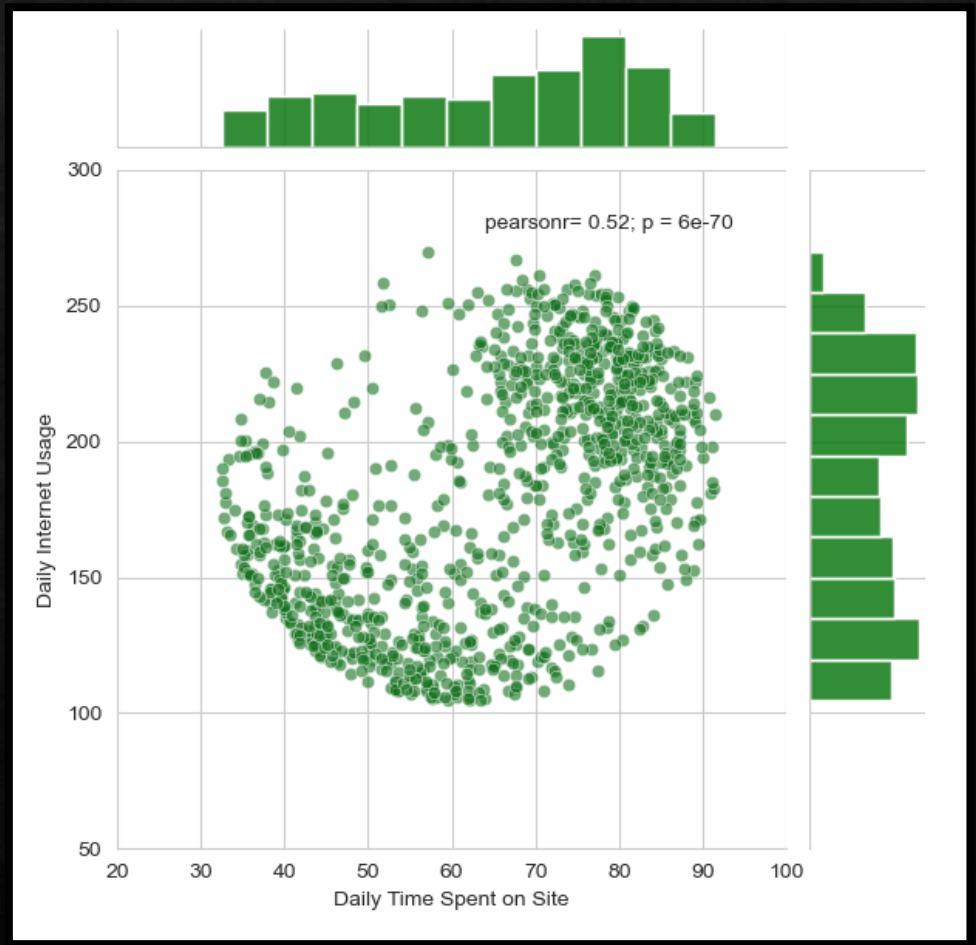
- The linear relationship between Age and Area Income is a small inverse relationship, which is evident from the Pearson correlation coefficient (pearsonr) and p-value.
- There is a high concentration of population aged between 20 and 40 with incomes ranging from 50,000 to 80,000.

AGE – DAILY TIME SPENT ON SITE JOINPLOT



- The linear relationship between Age and Area Income is a moderate inverse relationship, which is evident from the Pearson correlation coefficient (pearsonr) and p-value.
- There is a high concentration of population aged between 20 and 40, and daily time spent on the website ranges from 65 to 90.

DAILY TIME SPENT ON SITE – DAILY INTERNET USAGE JOINPLOT



- The linear relationship between Daily Time Spent on Site and Daily Internet Usage is evident from the Pearson correlation coefficient (pearsonr) and p-value.

- There is a high concentration of the population spending between 70 and 90 minutes daily on the website, and between 32 and 70 minutes daily on the website. Daily Internet Usage ranges from 175 to 250 minutes and from 110 to 175 minutes.

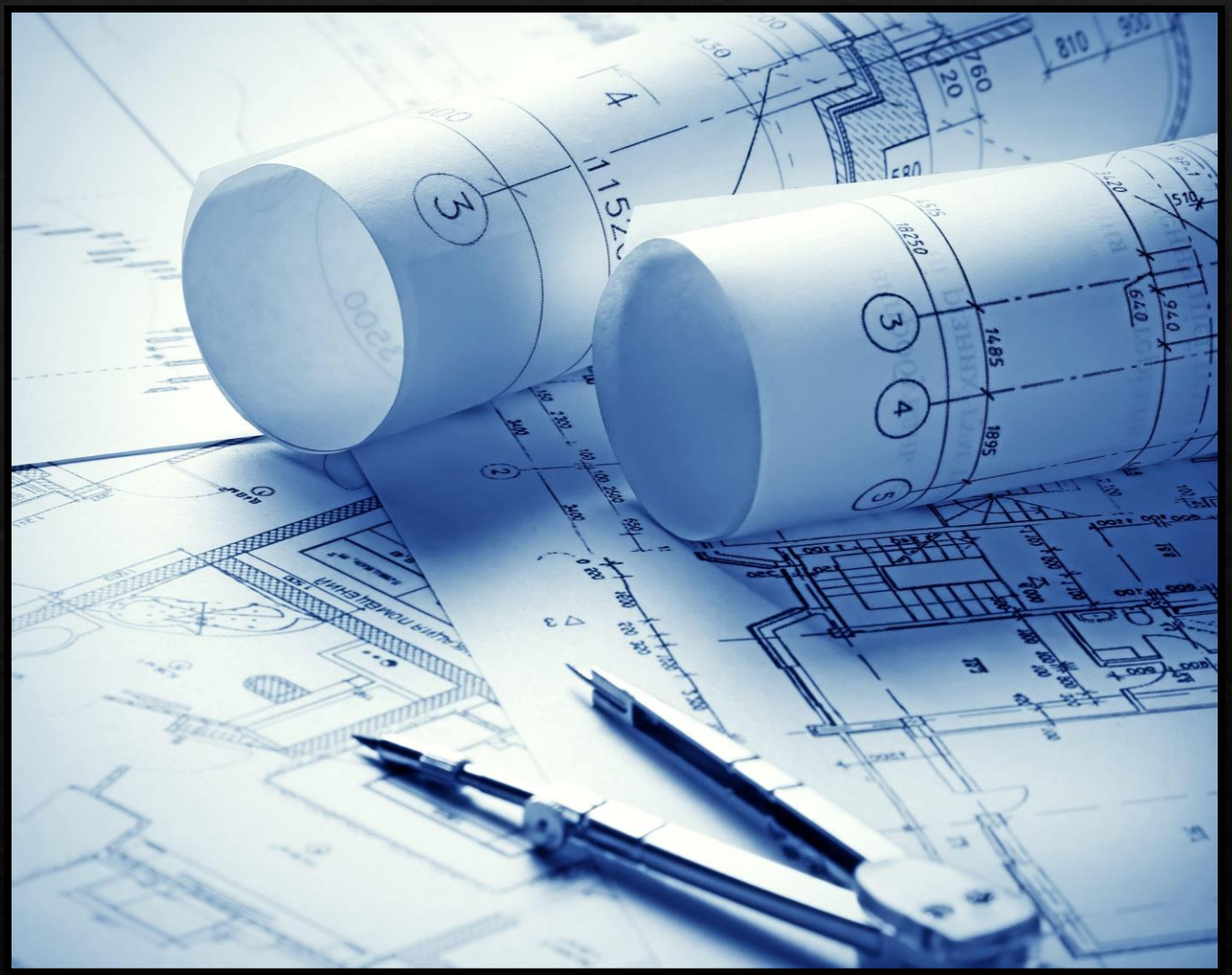
CLICK ON AD – ALL FEATURES SCATTER PLOT

- Most of the people who click on the ad spend less daily time on the site, typically between 20 and 60 minutes. They are generally aged between 30 and 50 years old, with intermediate incomes ranging from 40,000 to 60,000. Additionally, they do not use much daily internet, typically between 120 and 130 minutes.

HEAT MAP

- I have removed all string features, including the timestamp, because they add significant overhead due to their numerous categories.
- I eliminated the 'Male' feature due to its low correlation.

FEATURE ENGINEERING

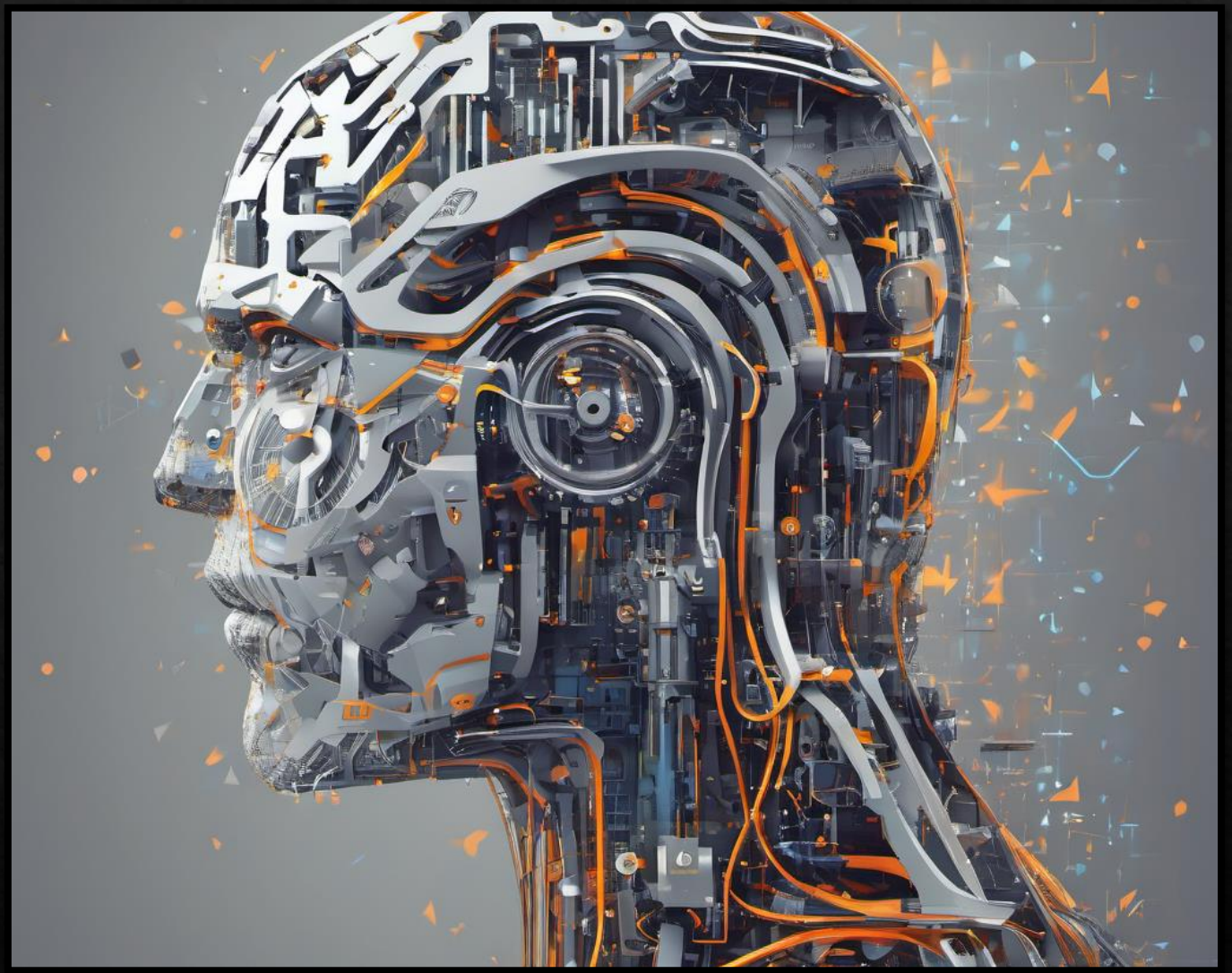


SCALING

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage
count	800.000000	800.000000	800.000000	800.000000
mean	0.153421	0.142527	0.935950	0.150078
std	0.112547	0.142683	0.090219	0.117650
min	0.000000	0.000000	0.000000	0.000000
25%	0.080686	0.052713	0.925459	0.075942
50%	0.138908	0.091671	0.957334	0.128471
75%	0.190828	0.186994	0.977295	0.191327
max	1.000000	1.000000	1.000000	1.000000

- We applied normalization to the feature to center its mean near zero and reduce outliers.
- After this we applied min-max scaling to make all the features within the same range.
- This is the best combination for increasing precision, but not recall. What is more important is that we don't want many false positives.

CHOOSING MODEL



LOGISTIC REGRESSION

I have chosen to use the Logistic Regression model with the following key parameters:

- **C=1.0:** Controls regularization strength, balancing model complexity and performance.
- **class_weight=None:** Assumes equal class weights, suitable for balanced **datasets.**
- **max_iter=100:** Ensures sufficient iterations for solver convergence.
- multi_class='ovr': Uses the one-vs-rest approach for multi-class classification.
- **penalty='l2':** Applies L2 regularization to prevent overfitting.
- **solver='liblinear':** Chosen for optimization, suitable for small datasets.
- **tol=0.0001:** Sets a precise convergence threshold.

K-NEAREST NEIGHBOUR CLASSIFIER

I performed a grid search to optimize the parameters of a K-Nearest Neighbors (KNN) classifier. The search included the following parameter grid:

- **metric**: Evaluated distance metrics 'euclidean', 'manhattan', and 'minkowski'.
- **weights**: Considered both 'uniform' and 'distance' weight options.
- **n_neighbors**: Tested neighbor counts from 1 to 30.
- Using cross-validation (cv=10) and accuracy as the scoring metric, the grid search identified the best parameters:
- **Best parameters**: {'metric': 'manhattan', 'n_neighbors': 1, 'weights': 'uniform'}
- **Best cross-validation accuracy**: 0.82

RANDOM FOREST CLASSIFIER

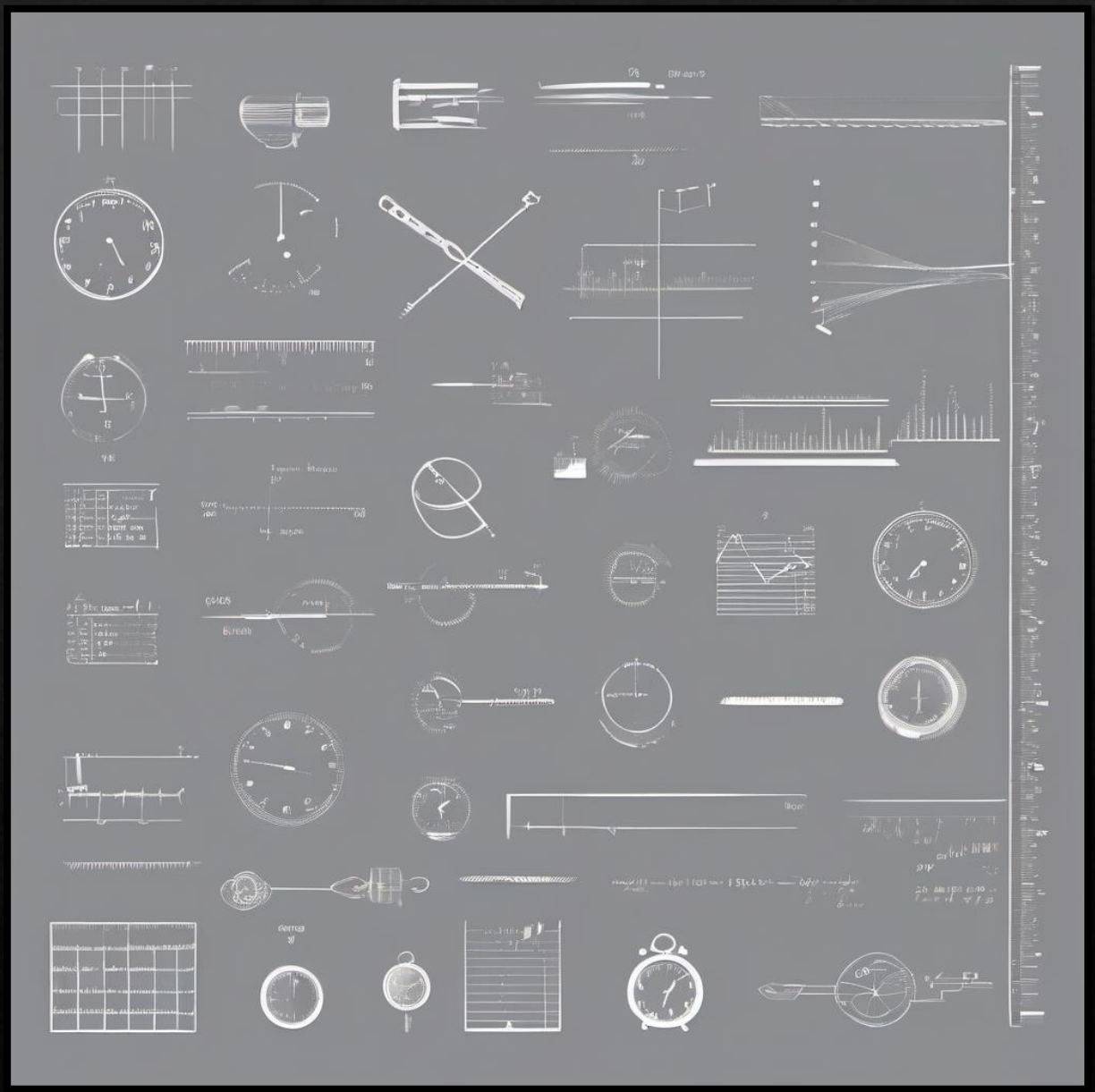
I conducted a grid search to optimize the parameters of a Random Forest classifier. The parameter grid included:

- **bootstrap**: Evaluated both True and False options to select samples for training each tree.
- **max_depth**: Tested None, 10, 20, and 30 for the maximum number of levels in the tree.
- **n_estimators**: Examined 100, 200, and 300 trees in the forest.
- **max_features**: Considered 'auto', 'sqrt', and 'log2' for the number of features to consider at every split.
- **min_samples_leaf**: Tested 1, 2, and 4 for the minimum number of samples required at each leaf node.
- **min_samples_split**: Evaluated 2, 5, and 10 for the minimum number of samples required to split a node.

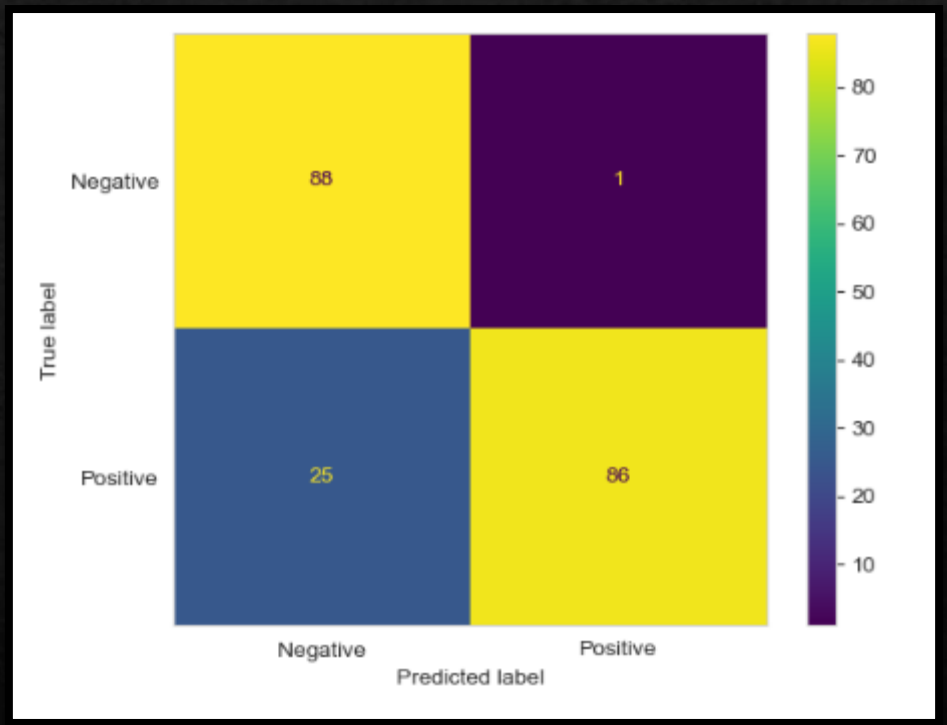
Using cross-validation (cv=10) and accuracy as the scoring metric, the grid search identified the best parameters:

- **Best parameters**: {'bootstrap': False, 'max_depth': 10, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 300}
- **Best cross-validation accuracy**: 0.97

EVALUATING MODEL

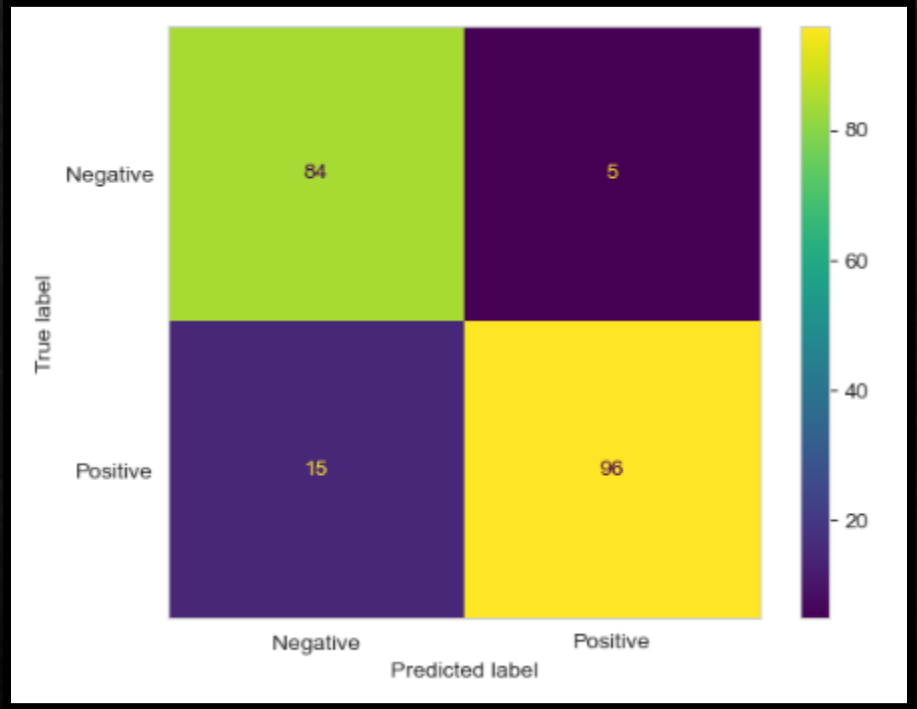


LOGISTIC REGRESSION



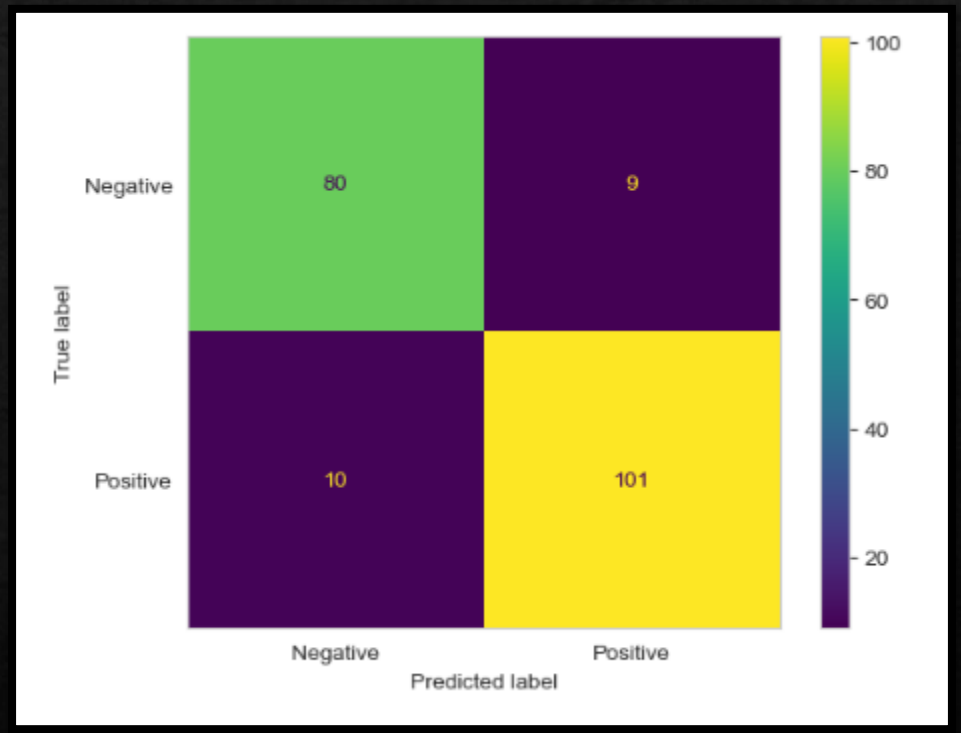

```
Precision : 0.99
Recall    : 0.77
F1 Score  : 0.87
```

K-NEAREST NEIGHBOUR CLASSIFIER




```
Precision : 0.95
Recall    : 0.86
F1 Score  : 0.91
```

RANDOM FOREST CLASSIFIER




```
Precision : 0.92
Recall    : 0.91
F1 Score  : 0.91
```

BEST MODEL

- We want the model with high precision, which in this case is Logistic Regression.

- If we want a model that balances between precision and recall, then we can choose the Random Forest Classifier.

Thank You