Advanced
Machine Learning

Table of Content

What will We Learn Today?

1. What is imbalance data
2. Handling imbalance data
3. Dimensionality reduction
4. Explainable AI

Profile

Professional

- Senior Data Analyst – Kompas (2021 – Present)
- Data Scientist – Rukita (2020 – 2021)
- Research Assistant Analyst – Ensterna (2017 – 2019)

Educational Background

- Nuclear Engineering – Universitas Gadjah Mada

Connect with me

- https://dataimpact.medium.com/
- https://www.linkedin.com/in/ariprabowo/
- https://github.com/densaiko

Ari Sulistiyo Prabowo

Imbalance data

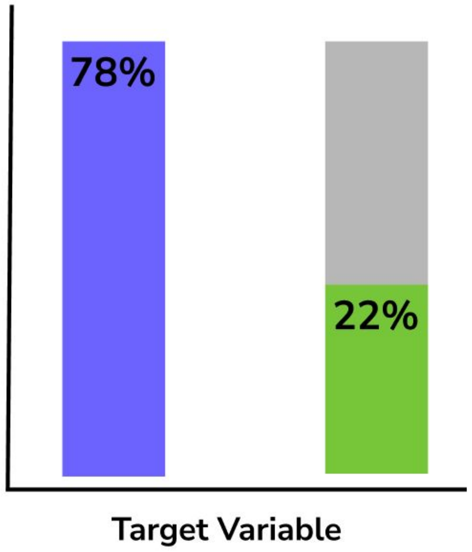
target variable yang memiliki jumlah data yang tidak seimbang 50:50

Imbalance data

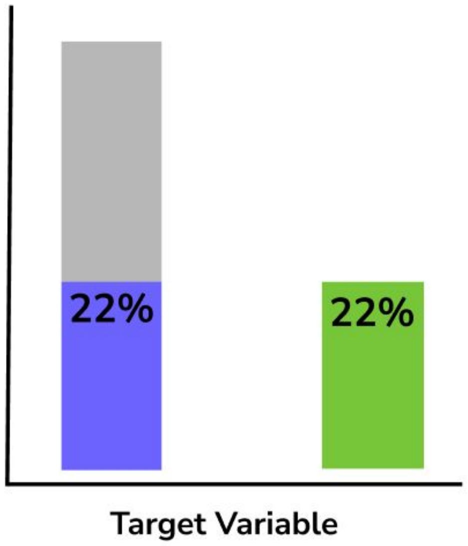
Terdapat dua hal dalam membuat data menjadi balance:

1. **Oversampling:** menambahkan data pada target variable yang sedikit
2. **Undersampling:** mengurangi data pada target variable yang banyak

Oversampling

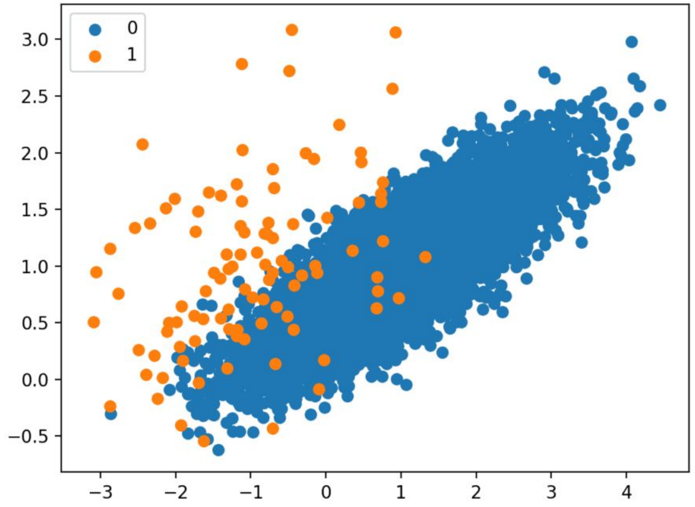


Undersampling

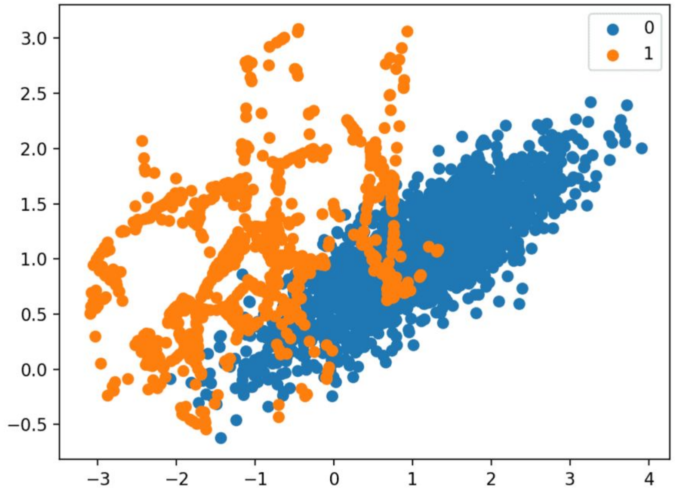


Imbalance data *(Smote)*

Dalam imbalance data, terutama oversampling, dilakukan penambahan data sintesis dengan menggunakan library **SMOTE**



Before Oversampling



After Oversampling

Imbalance data *(Smote)*

DigitalSkola
Uncover The World Of Digital Skills With Us

```
from imblearn.over_sampling import SMOTE #oversampling
```

```
# oversampling
sm = SMOTE(random_state=25, sampling_strategy=1) #sampling strategy 0.x to 1

# fit the sampling
X_train, y_train = sm.fit_sample(X_train, y_train)

y_train.value_counts()
```

```
Before smoting Counter({2: 5440, 1: 4703, 3: 2736, 0: 1638})
After smoting Counter({1: 5440, 3: 5440, 2: 5440, 0: 5440})
```

Dimensionality Reduction

Mengurangi dimensi suatu dataset

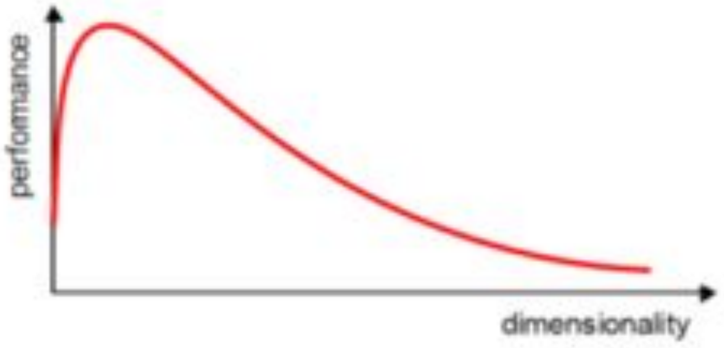
Dimensionality Reduction

Manfaat melakukan pengurangan dimensi:

- Mengurangi misleading data yang membuat akurasi model meningkat
- Mengurangi dimensi, mengurangi komputasi
- Mengurangi feature yang redundant

Terdapat **dua teknik** di dalam dimensionality reduction

- Feature selection
- Feature extraction



Hands on *(binary classification)*

Objective

X company would like to assess the employee to get a promotion. There are some criteria whether this employee can be promoted or not. Therefore, HR needs help from data scientist to create a machine learning model.

Target Variable

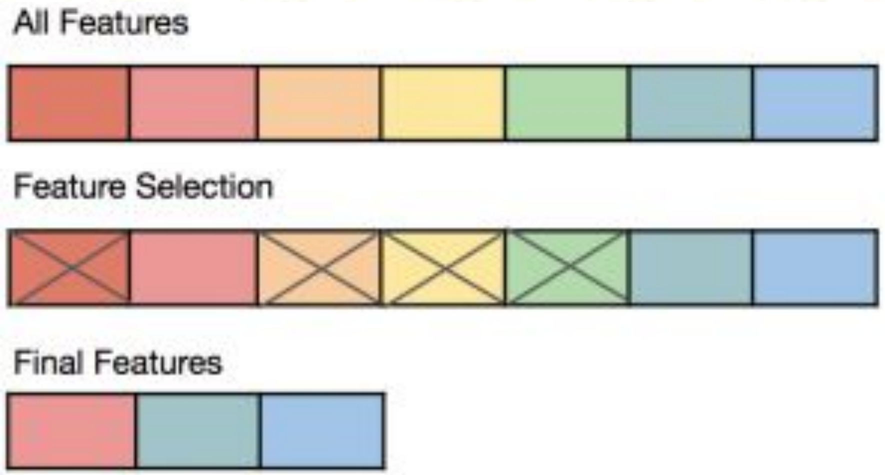
is_promoted

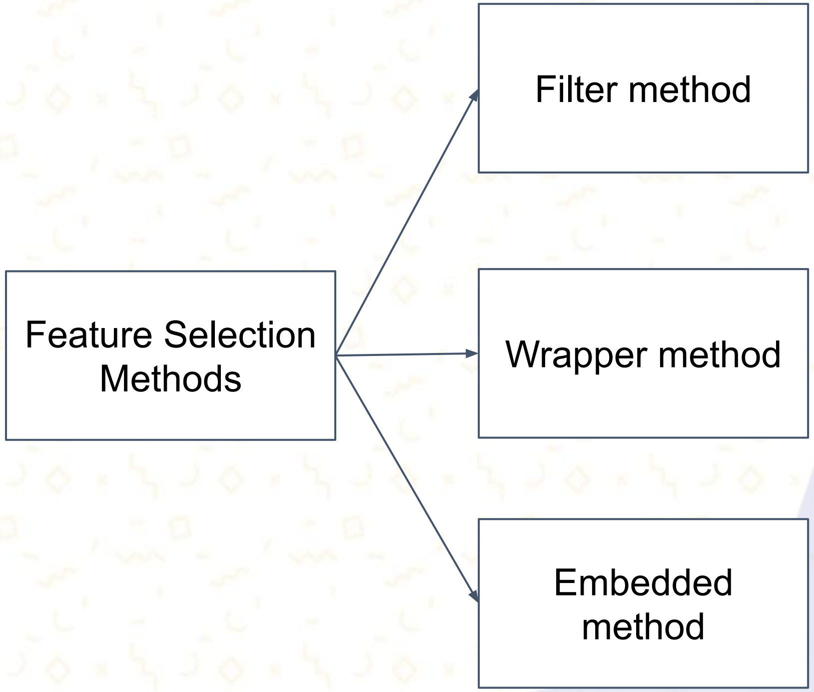
- 1 (promoted)
- 0 (not promoted)

Feature Selection

Feature selection adalah proses memilih subset dari fitur-fitur relevan dari seluruh fitur yang ada di dataset. Beberapa hal keuntungan feature selection:

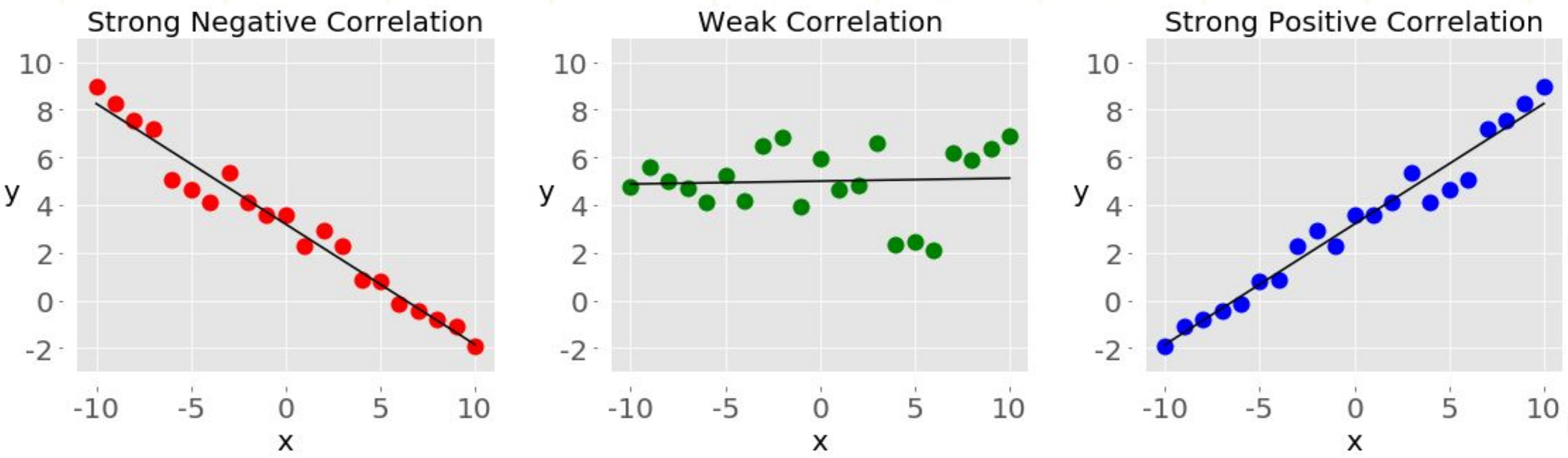
- Mengurangi waktu komputasi
- Mengurangi data yang tidak relevan
- Meningkatkan akurasi





Filter Method

Metode filter digunakan dengan melihat fitur-fitur yang memiliki korelasi yang tinggi.



Pada filter metode ini kita menggunakan metode statistik yaitu ANOVA yang digunakan untuk menganalisis variance untuk menentukan jika **rata-rata** dari lebih dari dua populasi adalah **sama**

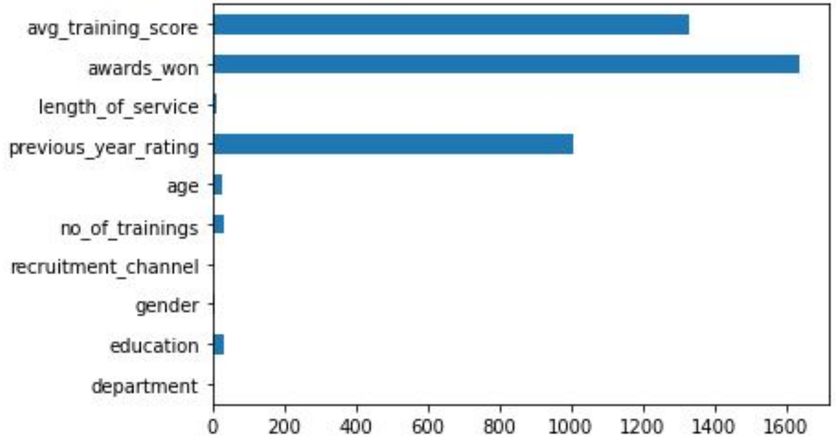
Filter Method

```python
from sklearn.feature_selection import SelectKBest, f_classif
```

```python
filter = SelectKBest(f_classif, k=5)
filter.fit(X_train, y_train)

X_train_new = filter.transform(X_train)
X_test_new = filter.transform(X_test)

print("Before feature selection", X_train.shape)
print("After feature selection", X_train_new.shape)
```

```
Before feature selection (37104, 10)
After feature selection (37104, 5)
```

```
Score of features [1.58485364e-01 3.26275563e+01 6.63837086e+00 1.47834343e+00
 3.00369928e+01 2.54122347e+01 1.00475625e+03 1.02120870e+01
 1.63478931e+03 1.33002169e+03]
```

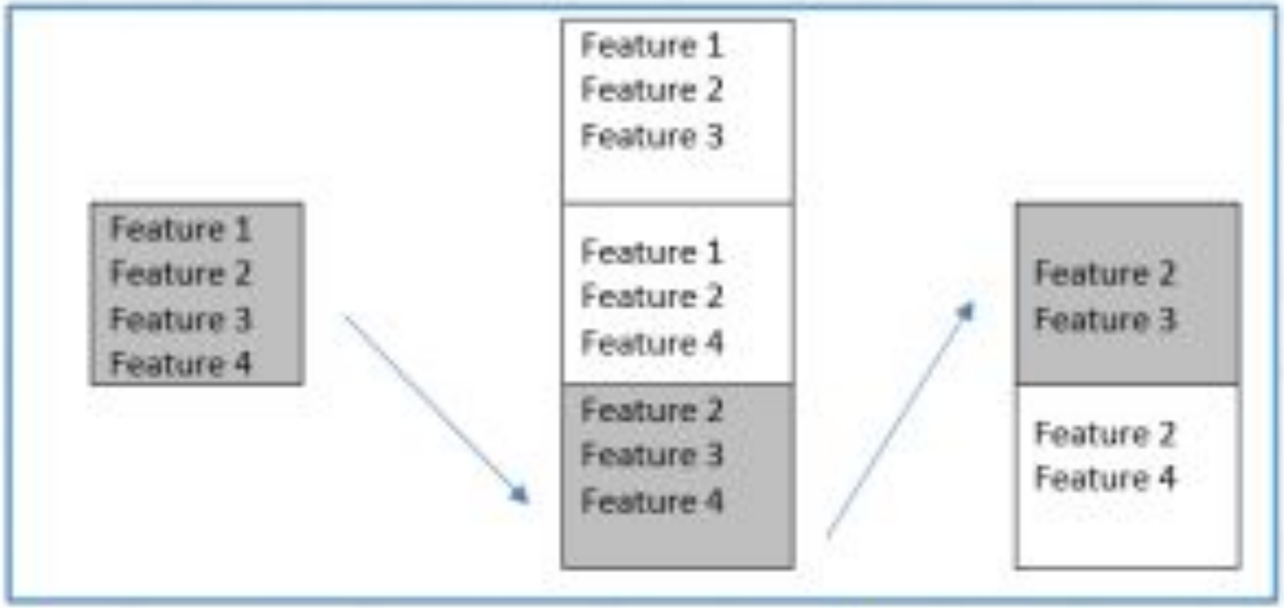


Features = 5
Selected Features = avg_training_score, awards_worn, previous_year_rating, education, no_of_trainings

Baseline ML (Logistic Regression)	91.79%
Logistic Regression + Anova	92.03%

Wrapper Method

Metode wrapper digunakan untuk menemukan kombinasi variable yang terbaik. Salah satu metode wrapper adalah RFE (Recursive Feature Elimination (RFE)



Wrapper Method

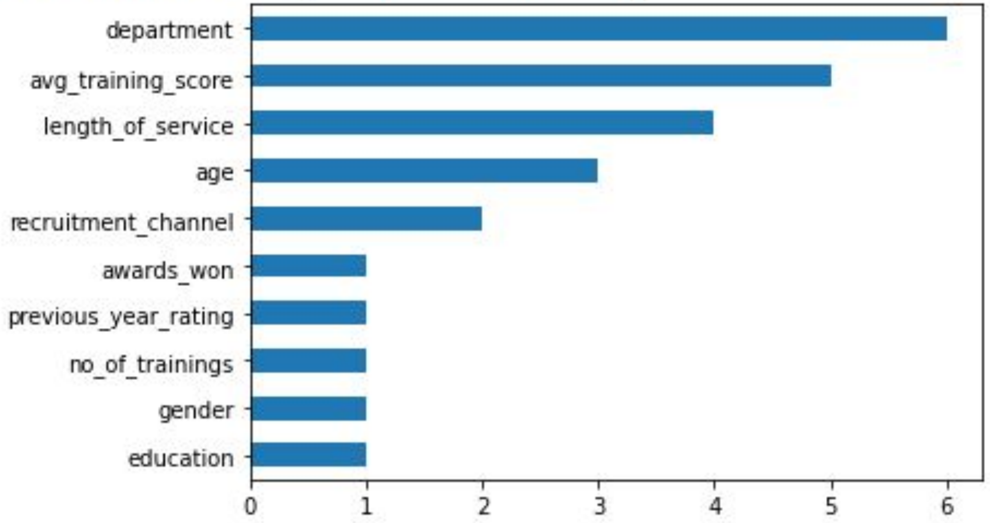
```python
from sklearn.feature_selection import RFE
```

```python
wrapper = RFE(clf, n_features_to_select=5)
wrapper.fit(X_train, y_train)

X_train_wrapper = wrapper.transform(X_train)
X_test_wrapper = wrapper.transform(X_test)
```

Features = 5
Selected Features = avg_training_score, awards_worn, previous_year_rating, education, no_of_trainings

Baseline ML (Logistic Regression)	91.79%
Logistic Regression + Anova	92.03%
Logistic Regression + RFE	91.83%

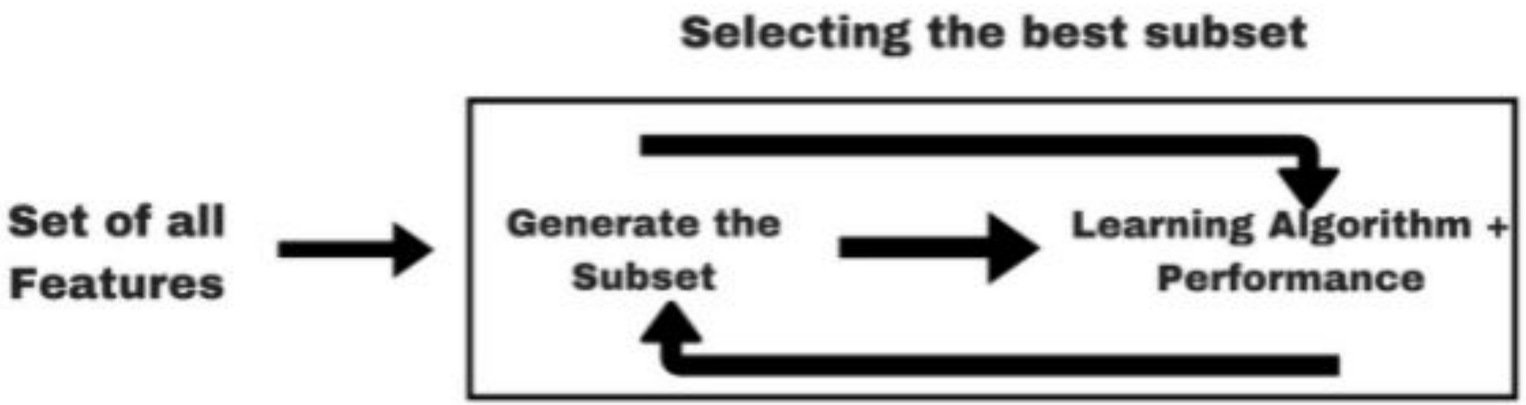
```
Before feature selection (37104, 10)
After feature selection (37104, 5)

Score of features [6 1 1 2 1 3 1 4 1 5]
```



DigitalSkola
Uncover The World Of Digital Skills With Us

Embedded Method

Metode embedded ini digunakan untuk memilih fitur-fitur mana aja yang digunakan dari hasil performa algoritma machine learning model



Embedded Method

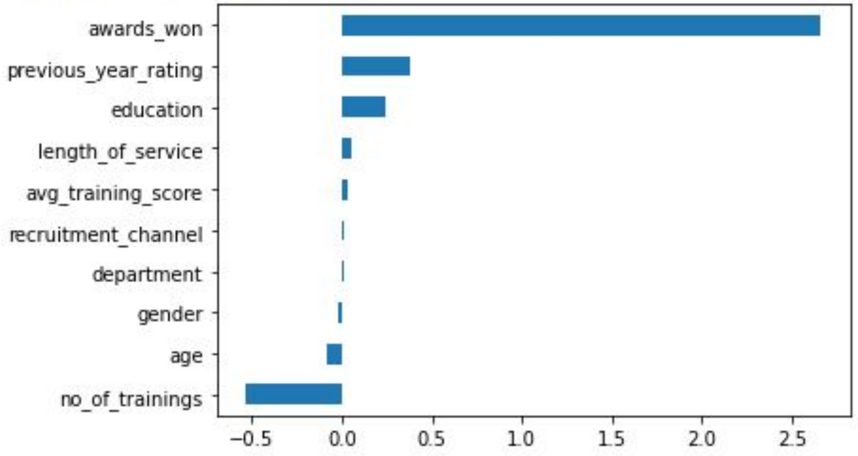
```
from sklearn.feature_selection import SelectFromModel
```

```
clf = LogisticRegression()
clf_feature = SelectFromModel(clf)

clf_feature.fit(X_train, y_train)

X_train_importance = clf_feature.transform(X_train)
X_test_importance = clf_feature.transform(X_test)
```

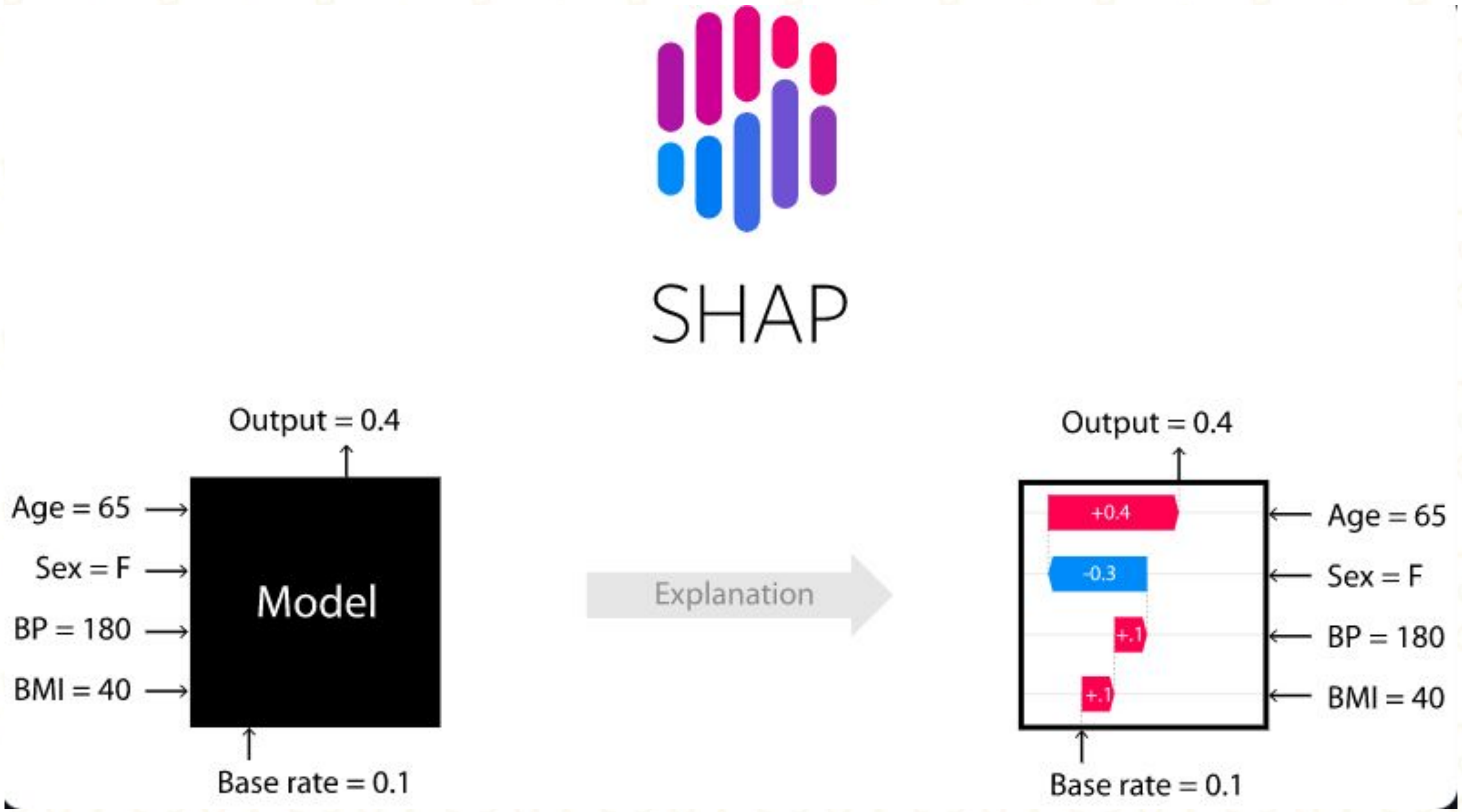
Features = 2
Selected Features = awards_worn, previous_year_rating

Baseline ML (Logistic Regression)	91.79%
Logistic Regression + Anova	92.03%
Logistic Regression + RFE	91.83%
Logistic Regression + Feature Importance	91.67%

```
Coef [ 0.01008519  0.2411234  -0.0189336   0.01141018 -0.53517601 -0.08410499
  0.37662892  0.05672754  2.66393455  0.02762389]
Treshold 0.4025748276559864
```



Explainable AI (BONUS)



Explainable AI (BONUS)

SHAP (SHapley Additive exPlanations) is a game theoretic approach **to explain the output of any machine learning model**. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions

Features **pushing the prediction higher** are shown in **red**, those **pushing the prediction lower** are in **blue**

Thank YOU