



# Generasi Gigih

Powered by



YAYASAN  
ANAK  
BANGSA  
BISA

goto

**STROKE PREDICTION ANALYS**  
DA\_HE1

SUPPORTED BY



Kalibrr



COMMUNITY  
PARTNER



# HELLO!



**This is member of our team**

## DA\_HE1

Salma Atsila Shabiyya (KM\_G2DA5239)  
Reyhan Naufal Rahman (KM\_G2DA4067)  
Rizka Luthfiani Farkhanah (KM\_G2DA4391)  
Azriel Rahmansyah Holy (P\_G2DA3055)  
Fatimah Azzahra (KM\_G2DA1217)



# Learning process that we get from #GenerasiGIGIH



Things we have to learn:

- Adaptability  
Help us to stay relevant and be more valuable where ever we work.
- Time Management  
We realize that good time management allow us to accomplish more in shorter period.
- Critical Thinking  
Help us to make an informed decision.
- Teamwork  
Learning with the group is a fascinating, make us more solid and cohesive

# Explanation on your Project background

- Our product focuses on one of the health problems that quite a lot of people experience, which is stroke. The product that will be made is a prediction of a person of condition on the potential of that person to have a stroke or not.
- With this product, people who have the potential for stroke will get recommendations such as continuing medical checks at the hospital directly as well as some tips on maintaining health and preventing stroke.
- everyone who wants to know if they have the potential to have a stroke.



# Problem Formulation on Stroke Analysis background

We use SMART statement as formulation of problem.

- **Specific** : Most people don't know that stroke can be prevented.
- **Measure** : Increase public awareness that stroke can be prevented by as much as 80%
- **Achievable** : Increase public awareness that stroke can be prevented.
- **Relevant** : Increasing awareness of stroke prevention in the community by implementing a healthy lifestyle and educating the public that stroke can be prevented
- **Time-bound** : Implement a healthy lifestyle in the community for 1 year



# Tech Use & Impact



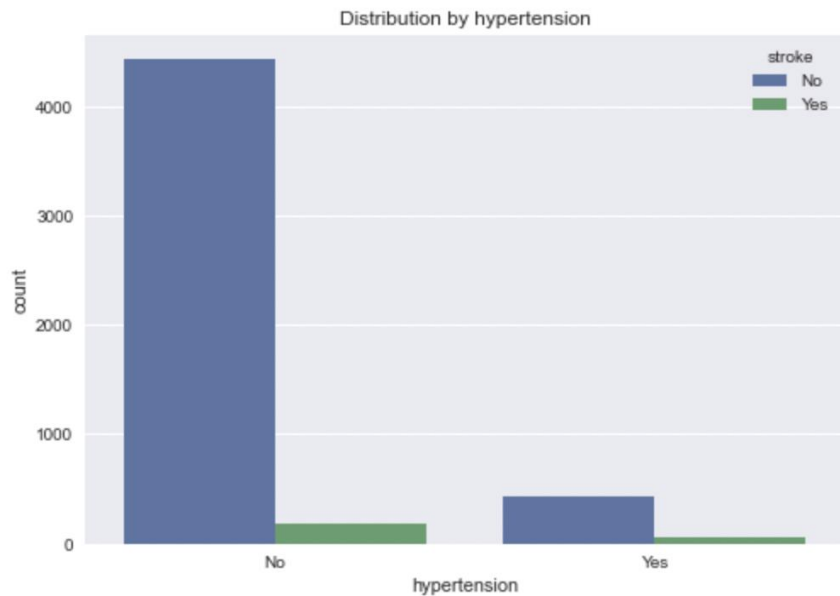
# Pandas, Numpy, and Matplotlib

## Data Visualization and EDA

Data visualization involves operating a huge amount of data and converts it into meaningful and knowledgeable visuals using various tools. For visualizing data we need the best software tools to handle various types of data in structured or unstructured format from different sources such as files, web API, databases, and many more.

In [33]:

```
count_bar_plot(data_eda, 'hypertension', 'stroke', 'Distribution by hypertension')
```

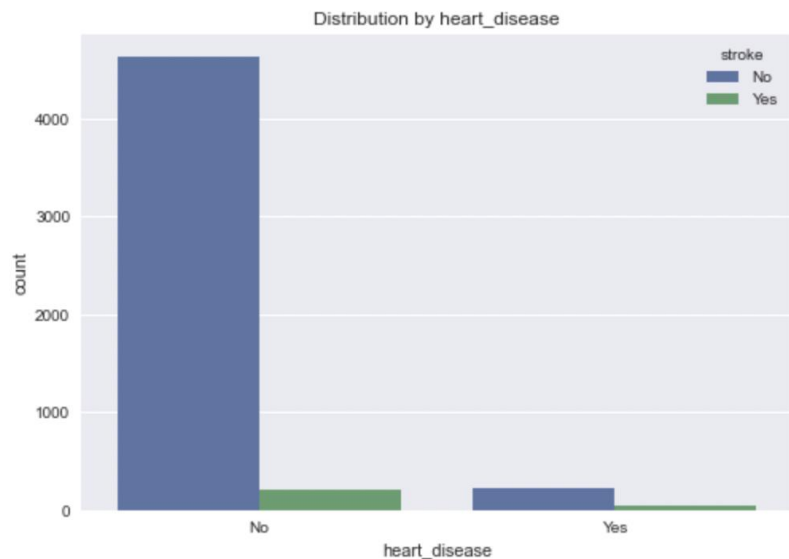


## Hypertension

- The patient who doesn't have hypertension represent 90%
- The patient who has hypertension represent 10%



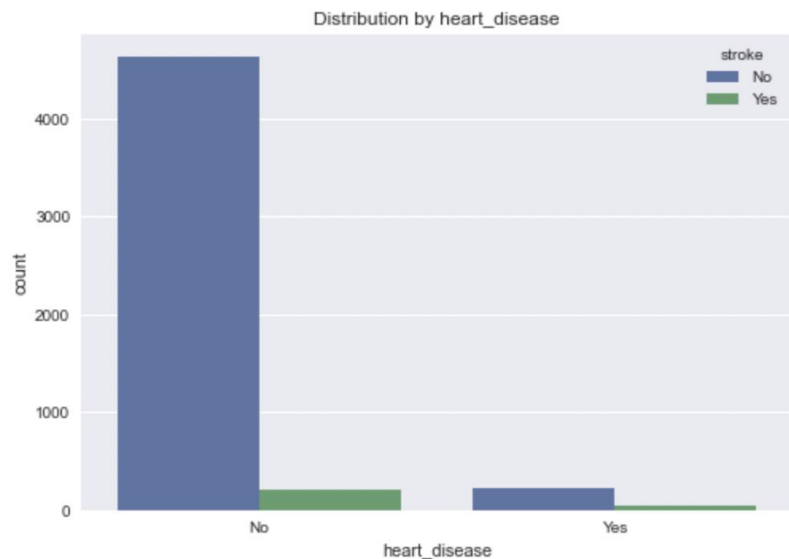
```
In [34]: count_bar_plot(data_eda, 'heart_disease', 'stroke', 'Distribution by heart_disease')
```



## Heart Disease

- The patient who doesn't have any heart diseases represent 94%
- The patient who has a heart disease represent 6%

```
In [34]: count_bar_plot(data_eda, 'heart_disease', 'stroke', 'Distribution by heart_disease')
```



## Herth Disease

- The patient who doesn't have any heart diseases represent 94%
- The patient who has a heart disease represent 6%

# Pyplot, Seaborn and Pandas

## Feature Engineering

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. In order to make machine learning work well on new tasks, it might be necessary to design and train better features. As you may know, a “feature” is any measurable input that can be used in a predictive model – it could be the color of an object or the sound of someone’s voice.

## Converting 'object' Column to 'integer'

```
In [45]: df.dtypes
```

```
Out[45]: gender          object
age          float64
hypertension    int64
heart_disease   int64
ever_married    object
work_type       object
Residence_type  object
avg_glucose_level float64
bmi            float64
smoking_status  object
stroke         int64
dtype: object
```

```
In [46]: df.head(2)
```

```
Out[46]:
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	formerly smoked	1
1	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	28.893237	never smoked	1

```
In [47]: df = pd.get_dummies(df)
df.head(2)
```

# Sklearn and Seaborn

## Modelling

The Model which provides the best result based on test findings is completed and deployed in the production environment whenever the desired result is achieved through proper testing as per the business needs. This concludes the process of Data Science Modelling.

## Random Forest Classifier Model

In [100...

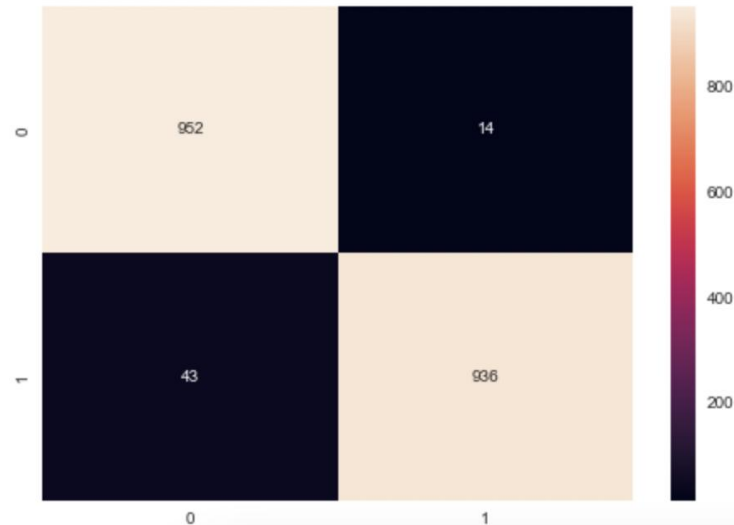
```
clf_rf = RandomForestClassifier(random_state=777)
clf_rf = clf_rf.fit(X_train,y_train)
y_pred_rf = clf_rf.predict(X_test)
acc = accuracy_score(y_test, y_pred_rf)
print('Testing-set Accuracy score is:', acc)
print('Training-set Accuracy score is:',accuracy_score(y_train,clf_rf.predict(X_train)))
cm = confusion_matrix(y_test, y_pred_rf)
sns.heatmap(cm, annot = True, fmt = "d")
```

Testing-set Accuracy score is: 0.970694087403599

Training-set Accuracy score is: 1.0

Out[100...

<AxesSubplot:>



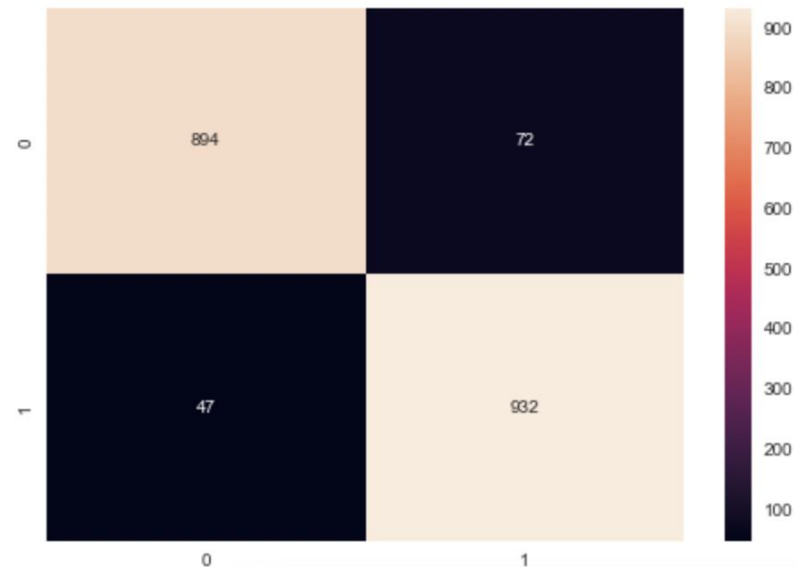
In [109...

```
dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)
dt_pred = dt.predict(X_test)
acc = accuracy_score(y_test, dt_pred)
print("Decision Tree accuracy score is :", acc)
cm = confusion_matrix(y_test, dt_pred)
sns.heatmap(cm, annot = True, fmt = "d")
```

Decision Tree accuracy score is : 0.9388174807197943

Out[109...

<AxesSubplot:>



In [107...

```
svc = SVC(random_state = 777)
svc.fit(X_train, y_train)
svc_pred = svc.predict(X_test)
acc = svc.score(X_test, y_test)
print("SVC Accuracy score is:", acc)
cm = confusion_matrix(y_test, svc_pred)
sns.heatmap(cm, annot = True, fmt = "d")
```

SVC Accuracy score is: 0.9598971722365038

Out[107...

<AxesSubplot:>





# Key Highlights [WIP]

SVC models

# Non-Technical Learning Takeaways

- Collaboration : How to keep being collaborative with each other's team member
- Better problem solving : We handled each problem with a proper method and created possible, practical and useful solutions
- Breakdown a problem : Team learned to make complex problem simpler
- In-Depth Understanding : We did some research to deepen our learning and for better project's results/outcomes
- Project Management : Team figured out how to effectively manage the project and each assignment
- Others : Critical thinking, self confidence, creativity, analytical thinking.



# Technical Learning Takeaways

- Collecting data using Kaggle
- Data Visualization using Google Data Studio
- Collaboration tool using Google Collaboration
- Data Preparation using Pandas Library
- Version control and collaboration using Github
- Modelling with Scikit Learn



**Thank you!**