MySkill Lion parcel

Portofolio - Short Class

Data Science

Owner: Agus Adiyanto

Build your skill and portfolio via myskill.id/bootcamp

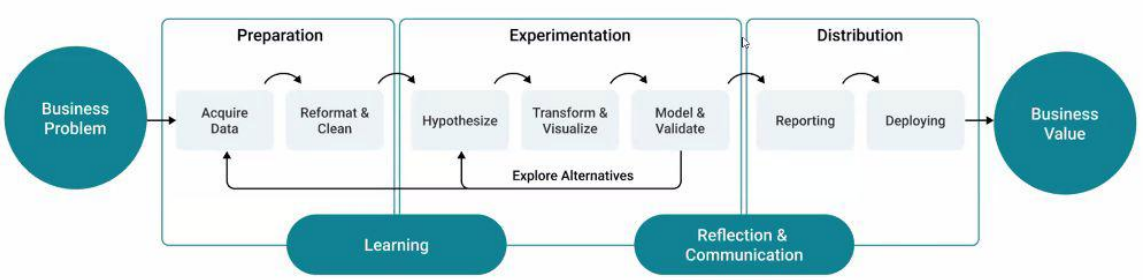
MySkill Lion parcel

Course Summary

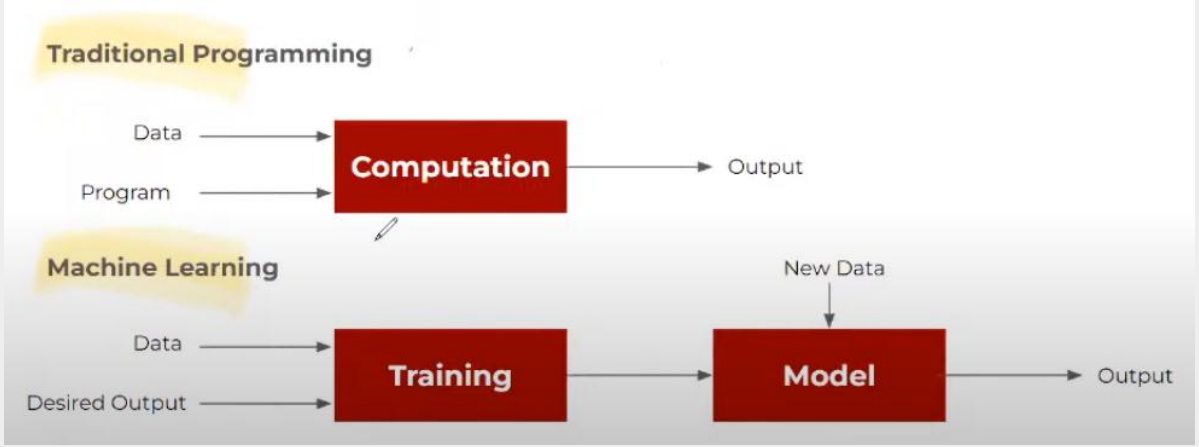
Poin Belajar	Rangkuman
The Anatomy of a Data Team	1. **Data Analyst:** insight, visualization, cleans, stores and organizes data. 2. **Data Engineer:** builds the data infrastructure (data warehouse). 3. **Data Scientist:** predicts what is going to happen, devise analytics approach, analyze data & identify insight, and build Machine Learning models
How To Become a Data Scientist	1. Get good at statistics and mathematics - Linear Algebra, - Calculus, - Statistics & Probability 2. Learn to code: Python, R, and SQL - Basic (data types, expression & variables, basic operator), - Programming Fundamentals (conditions & branching, loops, function, objects & classes) - Data Structure (list & tuples, sets, dictionaries) - Working with Data (pandas, matplotlib, machine learning) 3. Understand databases: MySQL, PostgreSQL, Amazon Redshift, Google Big Query 4. Master data munging (pandas), visualization (matplotlib), and reporting (tableau, power BI, looker studio)

MySkill Lion parcel

Course Summary

Poin Belajar	Rangkuman
How To Become a Data Scientist	5. Level up with big data (Spark, AWS, Google Cloud, etc) 6. Build your data science portfolio (Kaggle, GitHub, and LinkedIn) 7. Internship, bootcamp, and get a job 8. Follow and engage with the community
A Day in the Life of a Data Scientist	 Business Problem -> AcquireData -> Reformat & Clean -> Hypothesize -> Transform & Visualize -> Model & Validate -> Reporting -> Deploying -> Business Value.

Course Summary

Poin Belajar	Rangkuman
Machine Learning	Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behavior. Machine/computer must go through training based on a combination of data available to develop a foundation for further learning. 

MySkill Lion parcel

Task: Diabetes Prediction System with KNN Algorithm

DESCRIPTION AND DATASET :

Dataset terdiri dari variabel prediktif dan hasil yang menggambarkan apakah seseorang menderita diabetes atau tidak, Proyek mini ini bertujuan untuk membangun model pembelajaran mesin menggunakan metode pembelajaran terawasi dan algoritma k-Nearest Neighbor (KNN) dengan Python.

DATASET : https://bit.ly/DatasetSCDataMySkillxLionParcel

TOOLS REQUIRED :

- Notebook Code Editor
- Programming Language
- Python Package

RELATED DOCUMENTATION:

https://colab.research.google.com/
https://pandas.pydata.org/docs/user_guide/10min.html
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

MySkill Lion parcel

Practice Result

Diabetes Prediction System with KNN Algorithm.ipynb

File Edit View Insert Runtime Tools Help Last edited on May 8

+ Code + Text

Initialization and import libraries

```
[ ] # Import the required libraries

    import pandas as pd
    import time

    from sklearn.model_selection import train_test_split
    from sklearn.neighbors import KNeighborsClassifier
    from sklearn.metrics import confusion_matrix
```

```
[ ] # Read dataset diabetes.csv

    data = pd.read_csv('diabetes.csv',sep=',')
    display(data)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1

Import file diabetes.csv to Google Colab

Add library
Pandas, time, and sklearn
useful for connecting csv/dataset files, predictions, and data transformation.

Displays tables and columns
to display and separate data according to information (data type, column, row, value, etc.).

#RintisKarirImpian

MySkill Lion parcel

Practice Result

Diabetes Prediction System with KNN Algorithm.ipynb

File Edit View Insert Runtime Tools Help Last edited on May 8

+ Code + Text

```
[ ] # To display detailed information about the dataframe (number of rows of data,
    # column names along with the amount of data and data type)
    data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

data.info()

To display detailed information about the dataframe (number of rows of data, column names along with the amount of data and data type)

MySkill Lion parcel

Practice Result

Data Cleansing for missing data null

Looking for null data

```
[ ] data.isnull()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
763	False	False	False	False	False	False	False	False	False
764	False	False	False	False	False	False	False	False	False
765	False	False	False	False	False	False	False	False	False
766	False	False	False	False	False	False	False	False	False

Sum the null data

```
[ ] data.isnull().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

data.isnull() or data.isnull().sum()

Find out/look for empty data (NULL) with the aim of finding out whether the data is valid or not.

#RintisKarirImpian

MySkill Lion parcel

Practice Result

Delete the Outcome column data and make it data_x

```
[ ] data_x = data.drop('Outcome', axis=1)
    display(data_x)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33
...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63
764	2	122	70	27	0	36.8	0.340	27
765	5	121	72	23	112	26.2	0.245	30
766	1	126	60	0	0	30.1	0.349	47
767	1	93	70	31	0	30.4	0.315	23

768 rows × 8 columns

data_x = data.drop('Outcome', axis=1)
display(data_x)

Delete the Outcome column data and make it data_x

MySkill Lion parcel

Practice Result

Diabetes Prediction System with KNN Algorithm.ipynb

File Edit View Insert Runtime Tools Help Last edited on May 8

+ Code + Text

Display the Outcome column data and make it data_y

```
[ ] data_y = data['Outcome']
    display(data_y)
```

```
0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
```

data_y = data['Outcome']
display(data_y)

Display the Outcome column data and make it data_y

MySkill Lion parcel

Practice Result

Model Training

```
[ ] # Start time measurement for training
    start = time.time()

    # Split data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(data_x, data_y, test_size=0.33)

    # Create and train the KNN model
    model = KNeighborsClassifier(n_neighbors=3)
    model.fit(X_train, y_train)

    # End time measurement
    end = time.time()

    print(f"Model training in completed {end-start}")
```

```
Model training in completed 0.011564016342163086
```

Separating Original Data and Test Data

to determine the accuracy of the resulting data.

Training Data

Before the data is sent to the machine learning algorithm, the test data needs to be trained to make predictions.

The result is less than 1 second.

MySkill Lion parcel

Practice Result

```
[ ] # Demo predicted data

    new_data=pd.DataFrame({
        'Pregnancies':[1],
        'Glucose':[150],
        'BloodPressure':[72],
        'SkinThickness':[30],
        'Insulin':[0],
        'BMI':[33.6],
        'DiabetesPedigreeFunction':[0.351],
        # Fill in the age value for example 50 or 20
        'Age':[50]
    })

    # Show predicted data
    display(new_data)

    new_predict=model.predict(new_data)

    # If the array result is 0 then it is not diabetes, if the array result is 1 then it is diabetes
    display(new_predict)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	1	150	72	30	0	33.6	0.351	50

```
array([1])
```

Demo predicted data

Fill in the age value for example 50 or 20

If the array result is 0 then it is not diabetes, if the array result is 1 then it is diabetes

MySkill Lion parcel

Practice Result

Diabetes Prediction System with KNN Algorithm.ipynb

File Edit View Insert Runtime Tools Help Last edited on May 8

+ Code + Text

Model Evaluating (Make predictions on the test set)

```
[ ] # Model Evaluating

    y_pred = model.predict(X_test)

    # Calculate accuracy (use score method for KNeighborsClassifier)
    accuracy = model.score(X_test, y_test)
    print("Accuracy:", accuracy)

    # Calculate confusion matrix (correct spelling)
    confusion_result = confusion_matrix(y_test, y_pred)
    print("Confusion Matrix:\n", confusion_result)
```

```
Accuracy: 0.6889763779527559
Confusion Matrix:
 [[130  39]
 [ 40  45]]
```

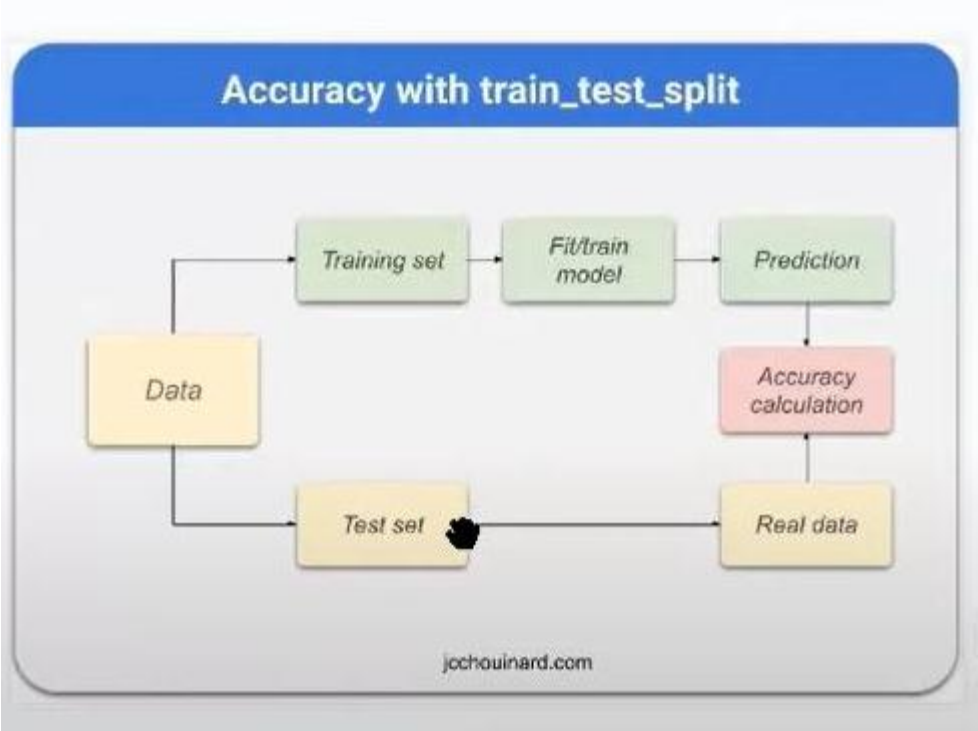
Check Accuracy

The results are
67% Accurate

#RintisKarirImpian

MySkill Lion parcel

Practice Result



How to Find Out the Generated Data Accurate Or Not

- Data is divided into two, namely original data and test data.
- Test data will later be tested and created model to predict what will happen happen in the future.
- The original data will later produce data real (now) used for take action now.
- Comparison of results from original data and data The test is used to determine how much accuracy of the data obtained.

MySkill Lion parcel

Follow me!

Instagram : @agus.adiyanto

Twitter : @gusadiyanto

LinkedIn : https://www.linkedin.com/in/agusadiyanto

Short Class Data Science and Data Analysis
by MySkill x Lion Parcel