Machine Learning Model Deployment
Model Tracking Using ML FLow

Group 1 Data Science B - MyEduSolve

Data Science B

KELOMPOK 1

Agung Wahyu P.

Informatics
Bhayangkara Jakarta Raya University

Fauziya Alya R.

Statistics
Sepuluh November Institute of Tecnology

Reno Darin K. P.

Electrical Engineering
Sepuluh November Institute of Tecnology

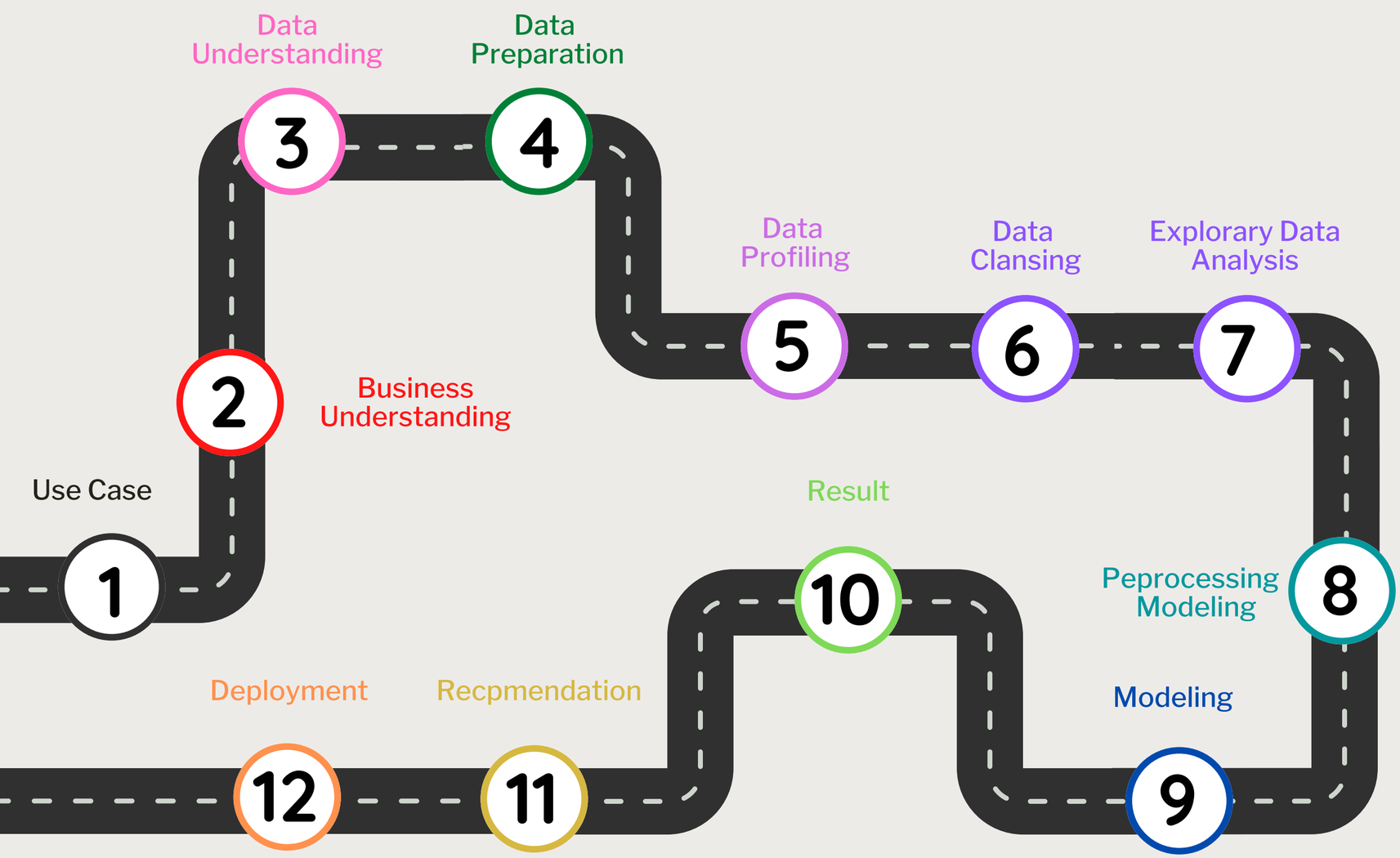
Putri Windasari

Mathematics
Diponegoro University

Valent Aderiandra

Information System
UPN "Veteran" East Java

WORKFLOW SUPERMARKET PREDICTION

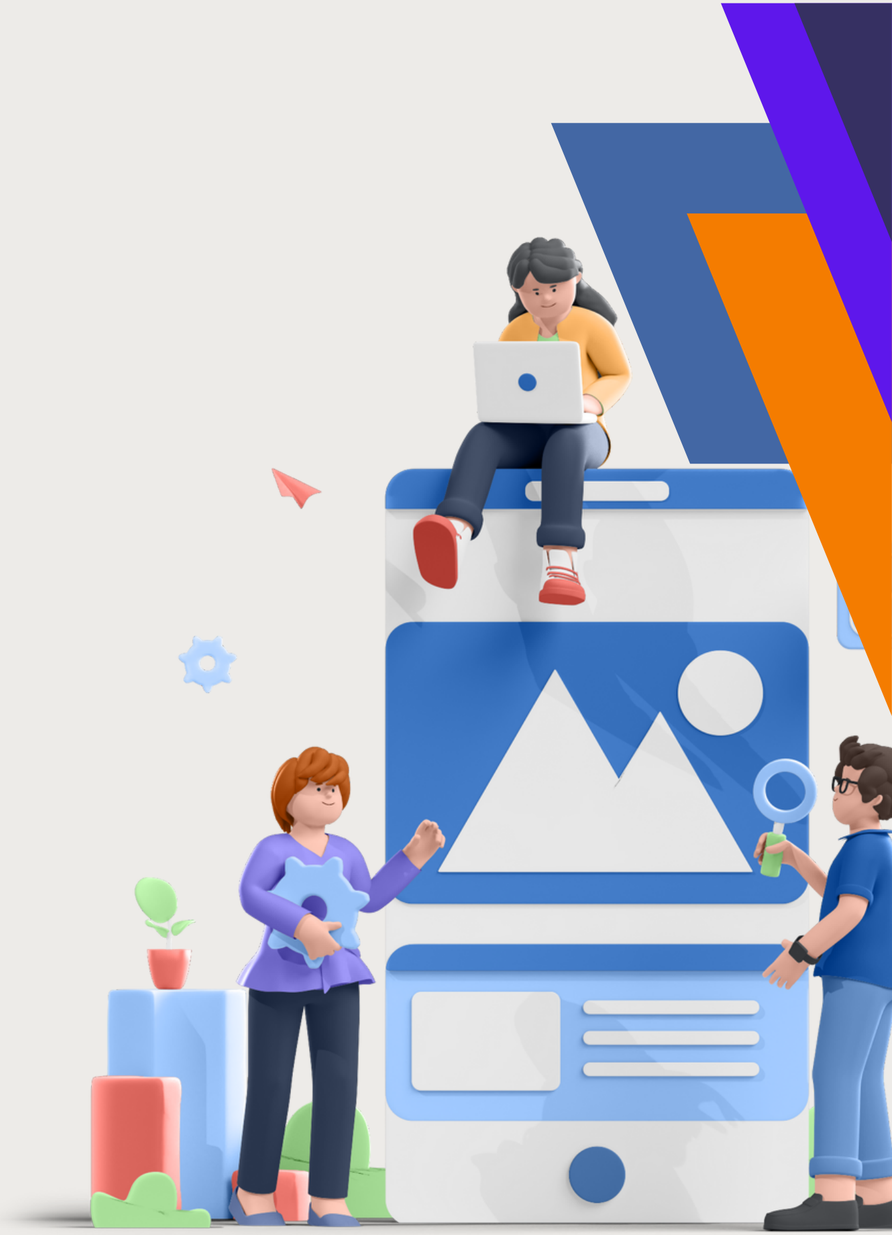


Use Case

Use Case Summary :

Objective Statement

- Get business insight about the average, maximum and minimum minimum visitor and buyer
- Get business insight about how many visitor in 30 day
- Get business insight about how many buyer in 30 day
- Get business insight about how many visitors buy more than once
- Get business insight about how to predict buyers based on visitors in one day using Machine Learning Simple Linear Regression
- Deployment using ML Flow



Challenges

- Do not know the time of data collection
- Don't know what the opening and closing hours are

Methodology / Analytic Technique

Descriptive analysis

- Describe the information such as,min/max value of each column, average, and the total count

Graph analysis

Using Machine Learning to predict

- Using Simple Linear Regression

Using ML Flow to Deployment

Business Benefit

- Gain insight to treat and keep customers based on segment
- Gain insight to improve the quality of company services so that customers remain loyal and gain more profit for the company
- Build machine learning using Simple Linear Regression
- Deployment using MLFlow

Expected Outcome

- Know about the average, maximum and minimum minimum visitor and buyer
- Know how many visitor in 30 day?
- Know how many buyer in 30 day?
- know how many visitors buy more than once?
- Know how to predict total buyers based on visitors in one day using Machine Learning Simple Linear Regression ?
- Know how to Deployment using MLFlow?

Business Understanding

Data is a sales data in supermarkets based on visitors and buyers within 30 days

- How about the average, maximum and minimum minimum visitor and buyer
- How many visitor in 30 day?
- How many buyer in 30 day?
- How many visitors buy more than once?
- How to predict buyers based on visitors in one day using Machine Learning Simple Linear Regression?
- How to Deployment using ML Flow?

Data Understanding

Source Data

- The dataset used is data from https://www.kaggle.com/datasets/reyanmatrika/supermarket-visitor-linear-regression
- 30 day sales data
- The raw data contains 30 rows and 2 columns.

Data Dictionary

- Visitor : total number of visitors
- Buyer : total number of visitors

Data Preparation

Code use

Python 3.9.13

Package

- Pandas,
- Numpy,
- Matplotlib,
- Seaborn,
- Scipy,
- Sklearn,
- and Warning

Data Profiling

Data profiling refers to the process of examining, analyzing, reviewing and summarizing data sets to gain insight into the quality of data.

Import Packages

In [1]:
```
# pip install mlflow
```

In [2]:
```
# pip install mlflow[extras]
```

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error, r2_score
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV

import warnings
warnings.filterwarnings('ignore')
```

Data Profiling

1.Load data set

```
In [4]: df = pd.read_csv("supermarket_buyer.csv")
```

2.Show top 5 data

```
In [5]: df.head()
```

Out[5]:

	Visitor	Buyer
0	34	32
1	38	36
2	34	31
3	40	38
4	30	29

3.Info data set

```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Visitor  30 non-null     int64
 1   Buyer    30 non-null     int64
dtypes: int64(2)
memory usage: 608.0 bytes
```

4.Check missing value

```
In [7]: df.isna().sum()
```

```
Out[7]: Visitor    0
        Buyer      0
        dtype: int64
```

Data Cleansing

The data is clean because there are no missing values and the data types are appropriate so there is no need for data cleansing

```
In [7]: df.isna().sum()

Out[7]: Visitor    0
        Buyer      0
        dtype: int64
```

Data is clean because there are no null values in any column

```
In [6]: df.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 30 entries, 0 to 29
        Data columns (total 2 columns):
         #   Column   Non-Null Count  Dtype
        ---  ------   --------------  -----
         0   Visitor  30 non-null     int64
         1   Buyer    30 non-null     int64
        dtypes: int64(2)
        memory usage: 608.0 bytes
```

Data types are appropriate

Exploratory Data Analysis (EDA)

In statistics, data analysis analysis is the approach of analyzing a data set to summarize its main features, often using graphical statistics and other methods of data visualization

Describe()

From the results of a descriptive analysis using 30 rows of data, some insights were obtained. The distribution of the data is still normal, this can be seen from the small standard deviation. The average buyer is 35 visitors with 33 purchases. During busy times, visitors are at most 42 visitors with a maximum of 38 purchases. There are 29 visitors and at least 29 buyers, so it can be concluded that it is most likely that during quiet times each visitor buys 1 item. When there are 38 visitors, the average purchase is 36. When there are 35 visitors, the average 33 buyers. When there are 32 visitors, the average buyer is 31.

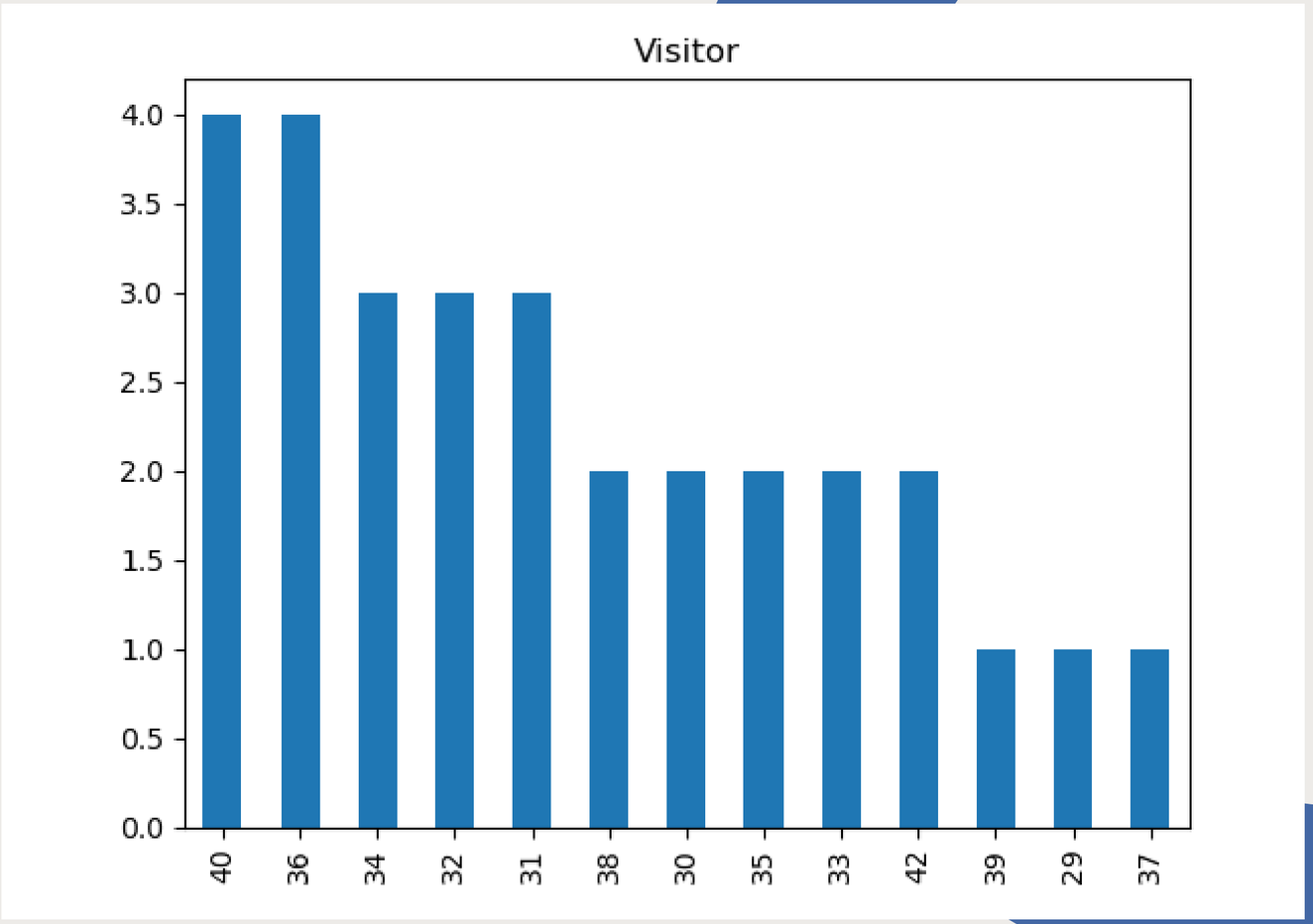
```
In [8]: df.describe()
```

Out[8]:

	Visitor	Buyer
count	30.000000	30.000000
mean	35.200000	33.400000
std	3.763711	2.647054
min	29.000000	29.000000
25%	32.000000	31.250000
50%	35.000000	33.000000
75%	38.000000	36.000000
max	42.000000	38.000000

Displays the number of visitors using a bar chart

From visitor data in the last month, most often (4 days) visitors came to the supermarket as many as 40 and 36 visitors. the possibility can happen because weekends in one month can happen 4x (Saturday Sunday). the loneliest day occurs when only 29 visitors visit one day, luckily it only happens for 1 day. it might be empty of visitors because it falls on a Monday and it's raining



Displays the number of buyer using a bar chart

From the last month's data, most often (6 days) there are 32 buyers in one day. maybe it could happen because it coincided with the weekend and there was a promotion. and the loneliest day for buyers occurred where there were only 29 buyers in one day.

Mode of visitor and buyer

Displays the mode of the visitor

- The Visitor mode consists of 36 visitors, which can be seen from the previous bar chart

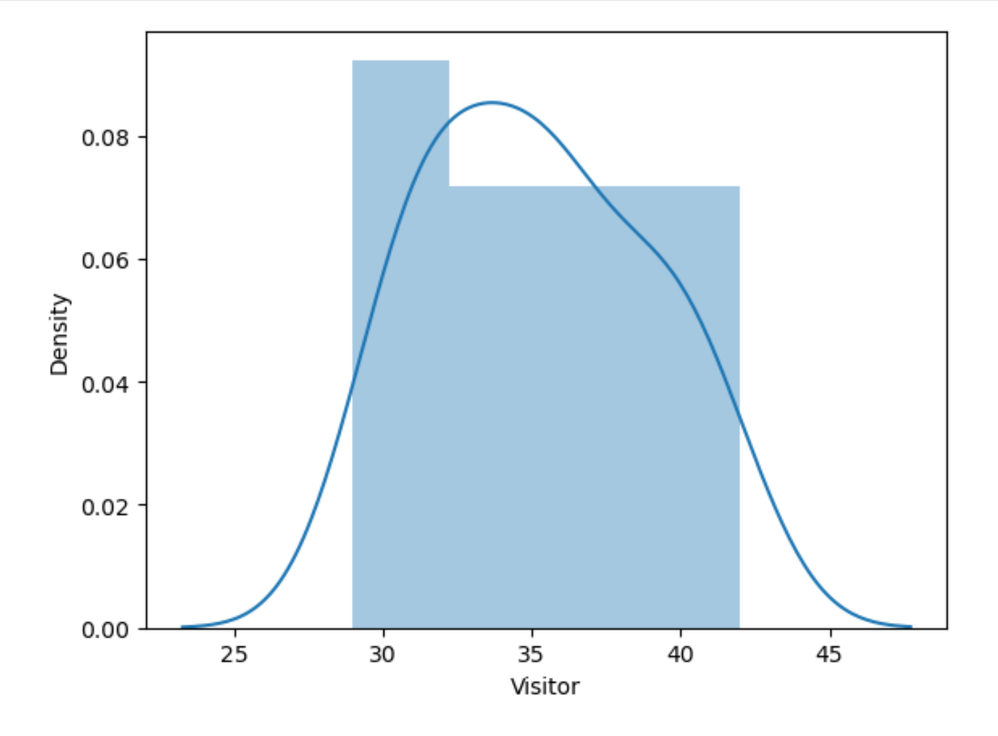
```
In [10]: df['Visitor'].mode().values[0]
Out[10]: 36
```

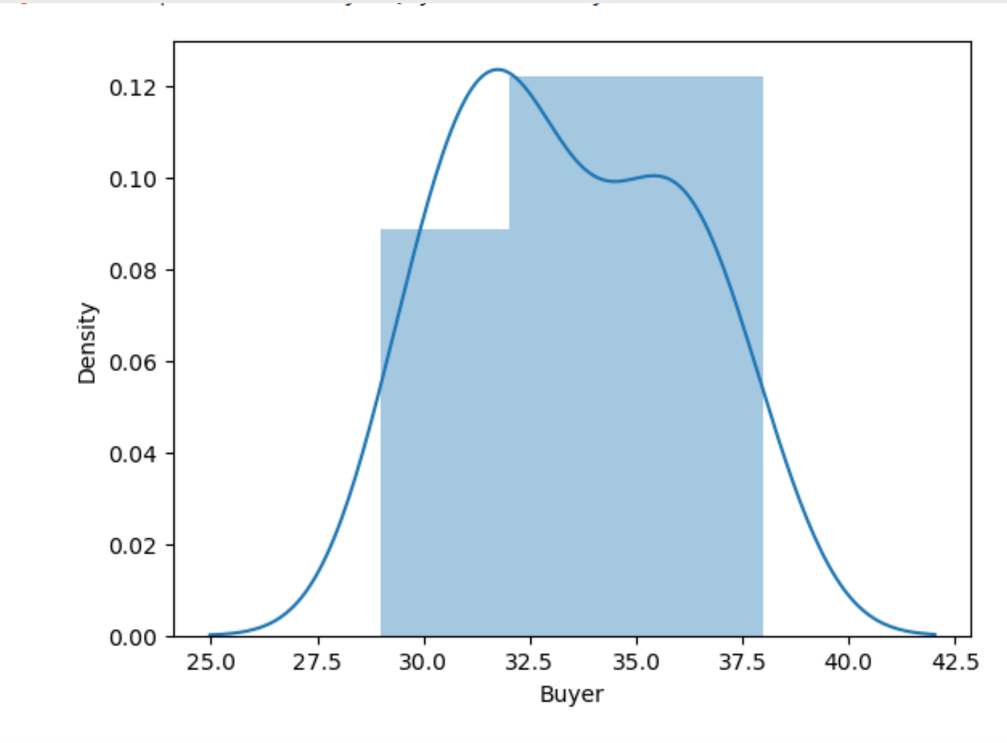
Displays the mode of the Buyer

- There are 32 modes of Buyer, so it can be concluded that most people buy a total of 32 items

```
In [12]: df['Buyer'].mode().values[0]
Out[12]: 32
```

Data distribution

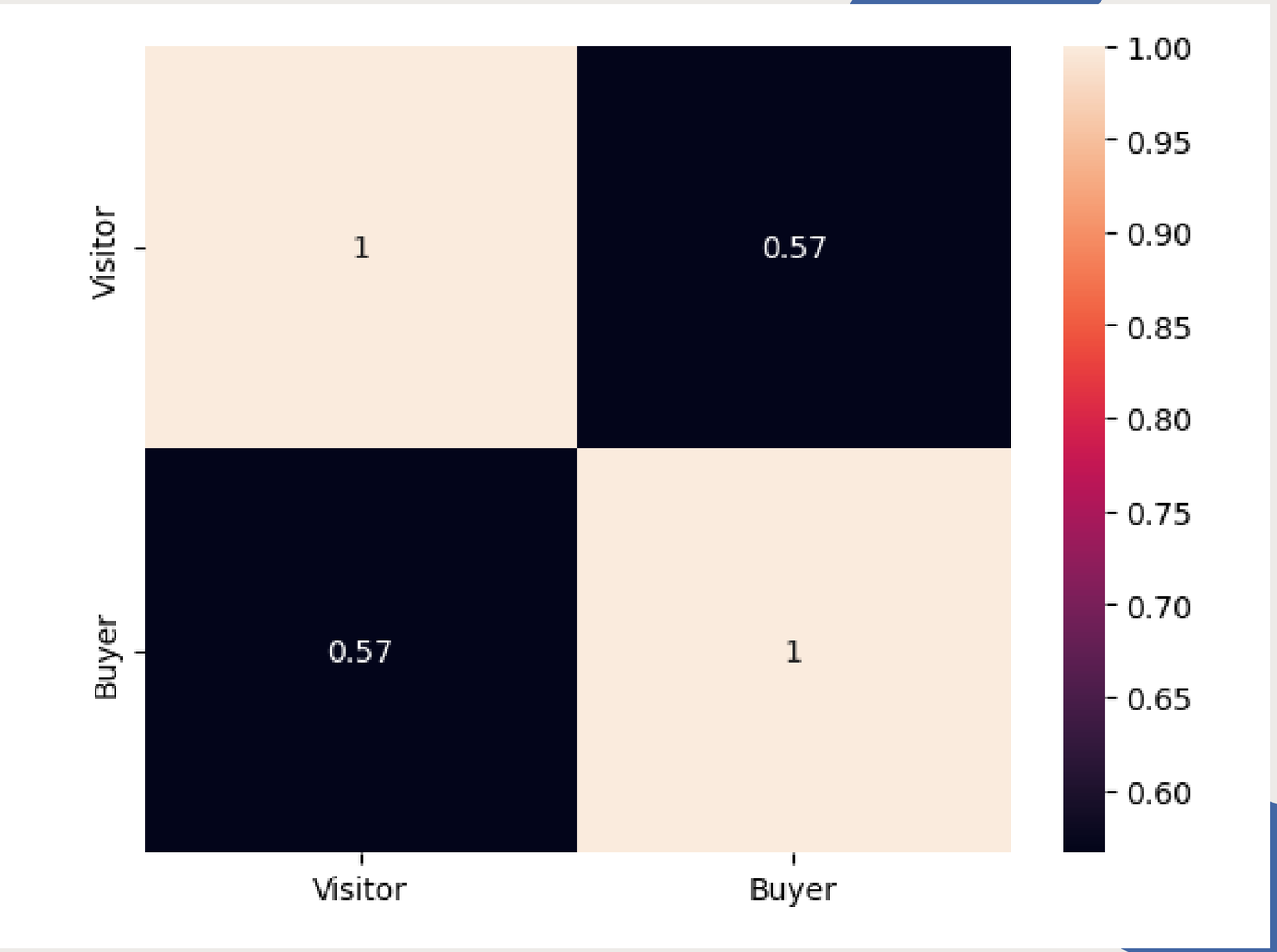




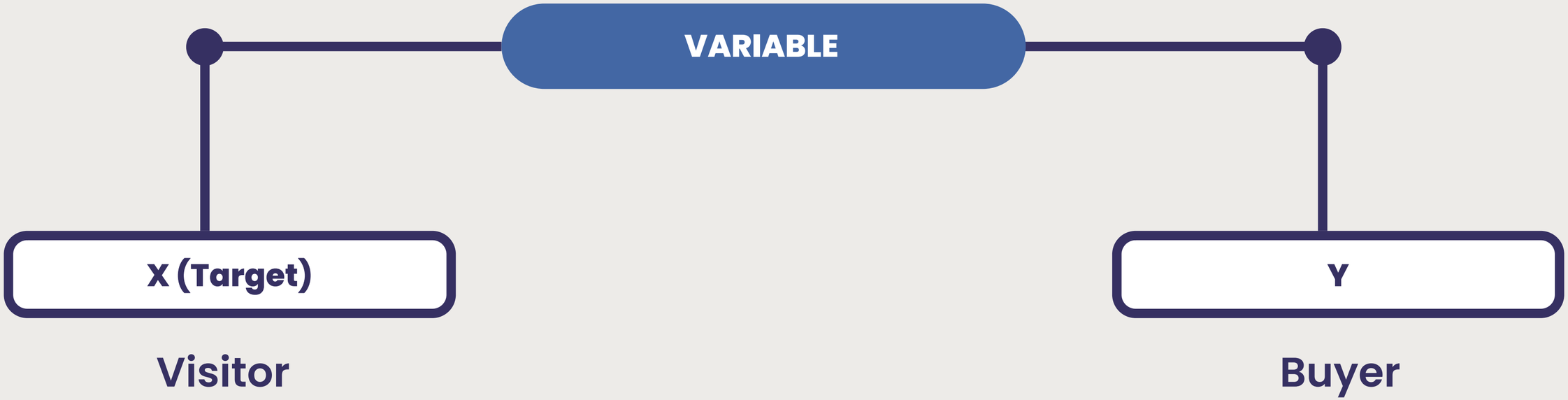
From the graph above it can be concluded that data distribution is not normal. because the mean, mode and median are not the same

Correlation

There is a strong relationship between total visitors and buyers. the more visitors the more buyers. with a correlation value on the heatmap of 0.57 where the figure is above 0.5



PREPROCESSING
MODELING



Here we do split training and test. Of the 30 data, we use 1/3 of the data to be trained and the rest to be tested

MACHINE LEARNING

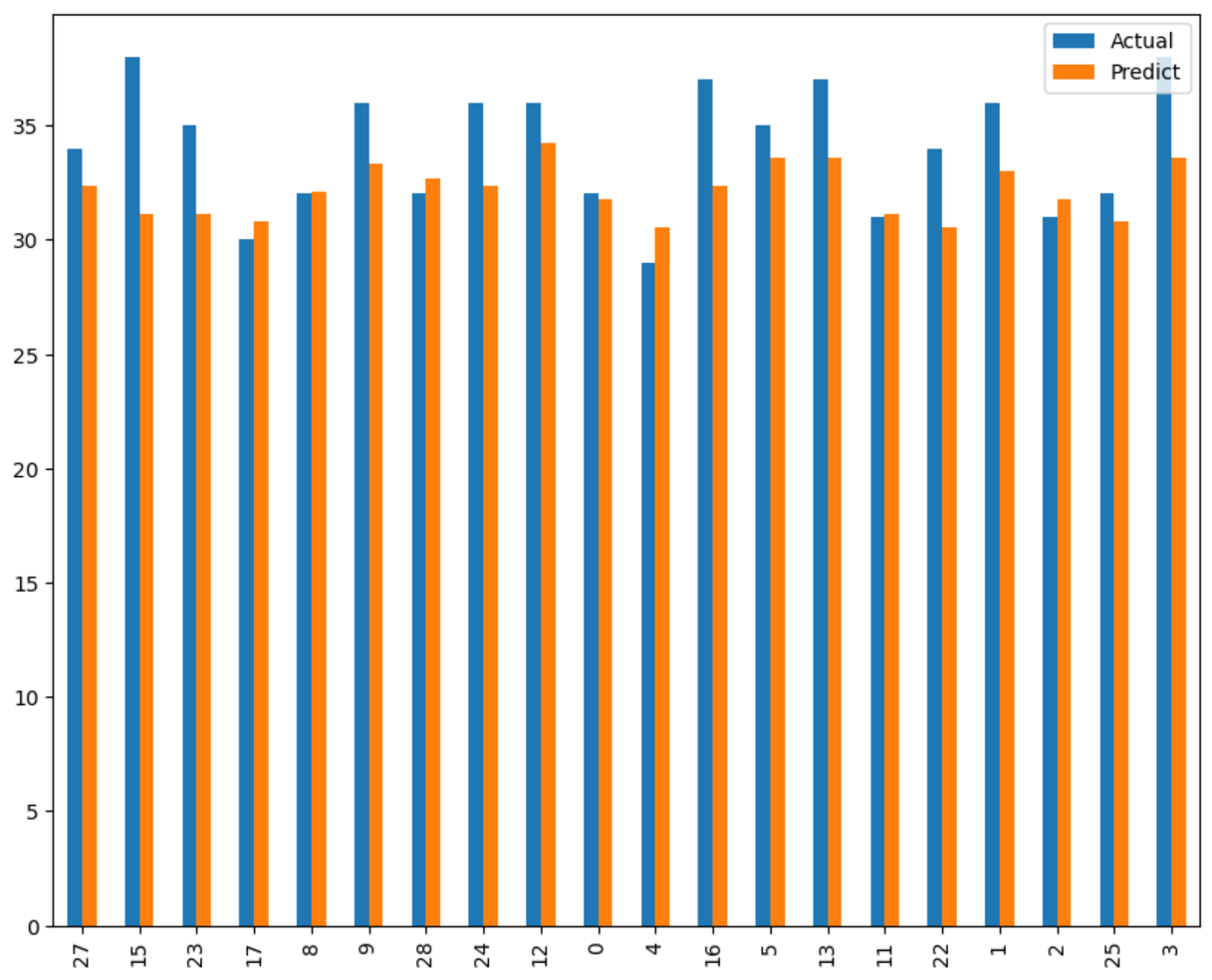
SIMPLE LINEAR REGRESSION MODELING

Simple Linear Regression

Definition

Simple linear regression is a regression model that estimates the relationship between one independent variable and one dependent variable using a straight line. Both variables should be quantitative.

Model Results

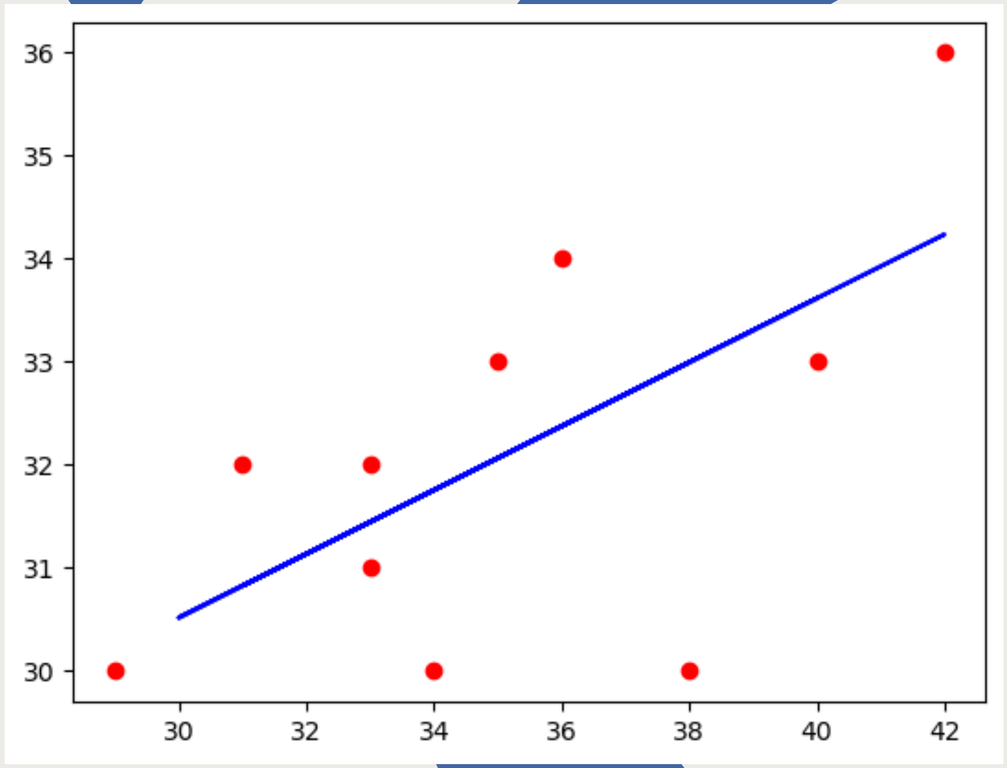


From the results of the barchart between the actual data and the predict data, it can be seen that the barchart of the predict data has not much difference from the barchart of the actual data. This shows that our model is good enough.

Model Results

The red dot on the scatter plot is the actual training data and the blue line is the predicted Y value line with X test values. The results of the scatter plot show that there is indeed a linear relationship between visitors and buyers. The more visitors, the more buyers. However, there are cases such as:

- In a day there are 42 visitors but a total of only 36 buyers, possibly because the goods run out or the visitors only look at the supermarket.
- Of the 29 visitors who bought, there were 30 possibilities because visitors bought more than 1 time.



Evaluate Model

RMSE

Root Mean Squared Error (RMSE) is one way to evaluate a linear regression model by measuring the accuracy of the predict results of a model.

MAE

MAE (Mean Absolute Error) is the average absolute difference between the actual (actual) value and the predicted (forecasting) value.

MAPE

Mean Absolute Percentage error (MAPE) is the absolute average percentage error.

R Squared

R squared is a number that ranges from 0 to 1 which indicates the magnitude of the combination of independent variables that jointly affect the value of the dependent variable. The R-squared value (R2) is used to assess how much influence certain independent variables have on the dependent variable.

Evaluate Model

RMSE	MAE	MAPE	R Squared
2.9032768988296604	2.30407177363699	6.482235882578176%	-0.17929580290702618

Simple Linear Regression Model With Cross Validation and Hyperparameter Tuning

We can combine simple linear regression models with hyperparameter tuning to obtain the best parameters to produce even better models.

Evaluate Model

RMSE	MAE	MAPE	R Squared
2.9032768988296604	2.30407177363699	6.482235882578176%	-0.17929580290702618

Comparing RMSE, MAE, MAPE and R Squared Without and With CV and Hyperparameter Tuning

RMSE	MAE	MAPE	R Squared
2.9032768988296604	2.30407177363699	6.482235882578176%	-0.1792958029070261 8
2.9032768988296604	2.30407177363699	6.482235882578176%	-0.1792958029070261 8

The accuracy value after using CV and hyperparameter tuning is not different from the model without CV and hyperparameter tuning. This can happen because the data is too little to do Machine learning and indeed the parameters used have been optimal before.

DEPLOYMENT

MLFlow Supermarket Prediction

Import Packages

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_squared_error, mean_absolute_error,
```

Load Dataset

```
# load dataset

df = pd.read_csv("Salary_Data.csv")
```

Preprocessing Models

```
X = df.drop(["Salary"],axis=1)
y = df["Salary"]

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 1/3, random_state = 42)

def eval_metrics(actual, pred):
    rmse = np.sqrt(mean_squared_error(actual, pred))
    mae = mean_absolute_error(actual, pred)
    mape = mean_absolute_percentage_error(actual, pred)
    r2 = r2_score(actual, pred)
    return rmse, mae, mape, r2
```

MLFlow Supermarket Prediction

Modeling

```python
# modeling

with mlflow.start_run():
        lr = LinearRegression (positive=False, normalize=True, n_jobs=None, fit_intercept=True, copy_X=True)
        lr.fit(X_train, y_train)

        y_pred = lr.predict(X_test)

        (rmse, mae, mape, r2) = eval_metrics(y_test, y_pred)

        print("  RMSE: %s" % rmse)
        print("  MAE: %s" % mae)
        print("  MAPE: %s" % mape)
        print("  R2: %s" % r2)


        mlflow.log_metric("rmse", rmse)
        mlflow.log_metric("mae", mae)
        mlflow.log_metric("mape", mape)
        mlflow.log_metric("r2", r2)


        tracking_url_type_store = urlparse(mlflow.get_tracking_uri()).scheme

        if tracking_url_type_store != "file":
            mlflow.sklearn.log_model(lr, "model", registered_model_name="Linear Regression")
        else:
            mlflow.sklearn.log_model(lr, "model")
```

Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\LENOVO\Machine Learning Coba> conda activate base
PS C:\Users\LENOVO\Machine Learning Coba> & C:/Users/LENOVO/ar
  RMSE: 6189.046433884711
  MAE: 5373.3795104073015
  MAPE: 0.08804898617394866
  R2: 0.9504877087663546
```

Result :

RMSE: 2.9032768988296604
MAE: 2.30407177363699
MAPE: 0.06482235882578176
R2: -0.1792958029070261 8

RESULT

Result

From the results of a descriptive analysis using 30 rows of data, some insights were obtained. The distribution of the data is still normal, this can be seen from the small standard deviation. **On average buyers are 35 visitors with 33 purchases**. **In busy times** , the **maximum number of visitors is 42 visitors with the most purchases are 38**. The **quietest visitors are 29 visitors and the minimum buyer is 29** so that **it can be concluded that most likely when it is quiet every visitor buys 1 goods**. When there were 38 visitors, the average purchase was 36. When there were 35 visitors, the average buyer was 33. When there were 32 visitors, the average buyer was 31.

Result

From visitor data in the last month, the most frequent (4 days) visitors came to the supermarket as many as 40 and 36 visitors. this might happen because weekends in one month can happen 4 times (Saturdays and Sundays). the loneliest day happened when one day only had 29 visitors, fortunately it only happened for 1 day. possibly there will be no visitors because it falls on Monday and it's raining

Result

from the last month's data, most often (6 days) there are 32 buyers in one day. it might happen because it coincides with weekend and there is a promotion. and the loneliest day of buyers occurred where there were only 29 buyers in one day.

can be seen here, the possibility of a visitor coming in one day to buy more than one item is 26% . this might happen because visitor's friends / family ask to buy the same item as the visitor

Result

From Visitor and Buyer distribution. the graph above it can be concluded that **data distribution is not normal**. **because the mean, mode and median are not the same**

From heatmap, there is a strong relationship of total visitors to buyers. **the more visitors the more buyers**. with a correlation value on the heatmap of **0.57** where the figure is above **0.5**

Result

We can get the RMSE of 2.9032768988296604. This RMSE is relatively small so that the model formed is good for predicting the data.

we can get MAE of 2.30407177363699. This RMSE is classified as small enough so that the model formed is good to predict the data.

Mape : absolute percentage of mean error is 0.06482235882578176 or 6% . This mape is classified as small enough so that the model formed is good for predicting the data.

This R2 score (-0.17929580290702618) means that the correlation is not too strong between visitors and buyers

Result

Results of RMSE, MAE, and MAPE from simple linear regression models without and with Cross Validation and Hyperparameter Tuning have values that are relatively small. This shows that the prediction error is also quite small so the model is classified as very good in terms of predicting the data.

When we compare with Cross model, the results obtained both before and after the results are the same with deployment code, we got same score like : RMSE: 2.9032768988296604,
MAE: 2.30407177363699, MAPE: 0.0648223588257817 6
R2: -0.1792958029070261 8

01

Recommendation

It's been good between visitors and total buyers, but needs to increase the value of existing goods because you can see there are still visitors just looking at them without buying

on weekend you can **multiply goods** because **buyers occur at that time**

on weekday you can **promote goods** and **discounts** for the number of buyers

DEPLOYMENT RESULT

MLFlow Modeling

MLFlow is a tool used to make it easier for Data Scientists so they don't have to worry about organizing models, recording each experiment, and facilitating the deployment process.

Deployment

Supermarket Prediction

Step by step MLFlow in VSC

Import Packages

load Daraset

Preprosecccing Model

Modelingl

MLFlow

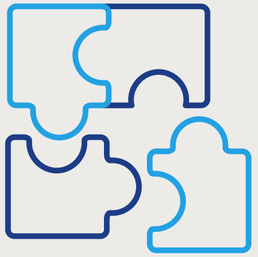
MLFlow features

MLFlow Tracking

Record and query experiments : code, data, config, and result.

MLFlow Project

Package data science code in a format to reproduce runs on any platform



MLFlow Models

Deploy machine learning models in diverse serving environment

Model Registry

Store, annotate, discover, and manage models in a central repository

There are several features in mlflow, but what we use is model tracking on rmse, mae, mape, and r2. Model tracking is a feature for recording experiments so that each experiment will be properly recorded in mlflow.

Deployment Result

ML Flow is a tool used to make it easier for Data Scientists so they don't have to worry about organizing models, recording each experiment, and facilitating the deployment process.

there are several features in mlflow, but what we use is model tracking on ***rmse, mae, mape, and r2 score***. Model tracking is a feature for recording experiments so that each experiment will be properly recorded in mlflow.

Model tracking result from mlflow is shown below:

Refresh | Compare | Delete | Download CSV | Created | All time

Columns | Only show differences | metrics.rmse < 1 and params.model = "tree" | Search | Filter | Clear

Showing 4 matching runs

							Metrics		
	Created	Duration	Run Name		Source	Version	Models	mae	mape	r2
	44 seconds ago	4.8s	honorable-g...	OVO	main.py	-	sklearn	2.304	0.065	-0.179
	2 days ago	4.7s	enthused-el...	OVO	main.py	-	sklearn	2.304	0.065	-0.179

Thank You

www.reallygreatsite.com