

LAPORAN PRAKTIKUM APLIKASI WEB

OLEH: MUHAMMAD EFFLIN RIZQALLAH LIMBONG (22537144007)

MODUL

TOPIK:

Streamlit Machine Learning (Model Regresi Linier) Pertemuan 8



```
78 // Trim the path and get the real path ($ _SERVER ['DOCUMENT_ROOT']) . ' ? _CAPTCHA & amp; t =' . ur
79 $ _SESSION ['_CAPTCHA'] ['config'] = serialize ($ captcha_config );
80
81 return array (
82     'code' => $ captcha_config ['code'],
83     'image_src' => $ image_src
84 );
85 }
86
87
88 // function_exists ('hex2rgb') ) {
89 // function hex2rgb ($ hex_str, $ return_string = false, $ separator = ',') {
90 // $ hex_str = preg_replace ('/[0-9A-Fa-f]/', '', $ hex_str); // Gets a proper hex string
91 // $ rgb_array = array ();
92 // if ( strlen ($ hex_str) == 6 ) {
93 //     $ color_val = hexdec ($ hex_str );
94 //     $ rgb_array ['r'] = 0xFF & ($ color_val >> 0x10 );
95 //     $ rgb_array ['g'] = 0xFF & ($ color_val >> 0x08 );
96 //     $ rgb_array ['b'] = 0xFF & $ color_val;
97 // } elseif ( strlen ($ hex_str) == 3 ) {
98 //     $ rgb_array ['r'] = hexdec (str_repeat (substr ($ hex_str, 0, 1), 2));
99 //     $ rgb_array ['g'] = hexdec (str_repeat (substr ($ hex_str, 1, 1), 2));
100 //     $ rgb_array ['b'] = hexdec (str_repeat (substr ($ hex_str, 2, 1), 2));
101 // } else {
102 //     return false;
103 // }
104 // return $ return_string ? implode ($ separator, $ rgb_array ) : $ rgb_array;
105 // }
106 // }
107 // }
108 // Draw the image
109 // if ( !isset ($ captcha_code )) {
110 //     $ captcha_code = $ captcha_config ['code'];
111 // }
```

TABLE OF CONTENTS

Week #4	1
A. Penjelasan Tugas Praktikum	3
B. Langkah-langkah dan Screenshot	4
C. Kendala yang Dialami	18
D. Kesimpulan	18

A. Penjelasan Tugas Praktikum

Streamlit merupakan library yang dimiliki Python yang bersifat open source. Streamlit sendiri dikeluarkan pada bulan Oktober tahun 2019. Ada beberapa Package yang terkandung pada Streamlit, diantaranya adalah Flask dan Django. Streamlit biasanya digunakan untuk membuat aplikasi web (Web Apps). Selain dari itu, Streamlit bersifat open source sehingga mudah untuk dibagikan ke pengguna-pengguna lain. Streamlit dapat memudahkan pengguna untuk mengubah data script menjadi aplikasi berbasis web yang interaktif.

Ketika Streamlit dieksekusi atau dijalankan, streamlit akan membuat server lokal. Aplikasi yang dibuat akan tampil di tab browser secara default. Tab browser tersebut adalah tempat dimana pengguna dapat membuat chart, text, widget, table, dan lain-lain.

Tugas praktikum pada pertemuan ini adalah eksplorasi data dan melatih model regresi linier pada aplikasi Streamlit.

B. Langkah-langkah dan Screenshot

A. LINEAR REGRESSION STREAMLIT

1. Kode Program Linear_Regression.ipynb:

```
# -*- coding: utf-8 -*-

"""Linear Regression.ipynb

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/11l4qvG96rI004VqoTUMejS4igkrLv2D7

# Very Simple Regression

## 1) IMPORT LIBRARIES AND DATASETS

"""
```

```
# Import Pkgs

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt


from google.colab import files

files.upload()


# read the csv file

student_marks_data =
pd.read_csv('student_marks_data.csv')


student_marks_data

student_marks_data.head()

student_marks_data.tail(8)
```

```

student_marks_data=
student_marks_data[student_marks_data['Marks'] >=
0]

# Check the minimum marks

student_marks_data['Marks'].min()

student_marks_data

"""## 2) Exploratory Data Analysis (EDA) and
Visualizations"""

# check if there are any Null values

student_marks_data.isnull().sum()

# Check the dataframe info

student_marks_data.info()

```

```

# Statistical summary of the dataframe

student_marks_data.describe()


# number of hours studied corresponding to
students with maximum marks

max
student_marks_data[student_marks_data['Marks']
student_marks_data['Marks'].max()]

max

# number of hours studied corresponding to
students with minimum marks

min
student_marks_data[student_marks_data['Marks']
student_marks_data['Marks'].min()]

min

# Histogram Plotting (Data Distribution)

```

```

student_marks_data.hist(bins    =   30,    figsize    =
(20,10), color = 'r')

# Plot Pairplot (Variables Relationship)

sns.pairplot(student_marks_data)


# Correlation Matrix

corr_matrix = student_marks_data.corr()

sns.heatmap(corr_matrix, annot = True)

plt.show()


# Regression Plot with Seaborn (straight line fit
between "Marks" and "Hours Studied")

sns.regplot(x='Hours_Studied',y='Marks',data=student_marks_data)


"""## 3) TRAINING AND TESTING DATA"""

X = student_marks_data[['Hours_Studied']]

```



```
y = student_marks_data[['Marks']]

X

y

X.shape

y.shape

X = np.array(X)

y = np.array(y)

X.shape

# split the data into test and train sets

from sklearn.model_selection import
train_test_split
```

```
X_train, X_test, y_train, y_test =  
train_test_split(X, y, test_size = 0.25,  
random_state=72)  
  
X_train.shape  
  
X_test.shape  
  
print(X)  
  
# We can see that data have been shuffled by  
"train_test_split"  
  
X_train  
  
"""## 4) Train a Linear Regression Model"""  
  
# using linear regression model  
  
from sklearn.linear_model import LinearRegression
```

```

from sklearn.metrics import mean_squared_error

lr = LinearRegression(fit_intercept = True) # Fit
intercept is the "b" parameter ( $y = b + mx$ )

lr.fit(X_train, y_train)

# Checking the accuracy

y_pred = lr.predict(X_test)

mse = mean_squared_error(y_test, y_pred)

mse

print('Linear Model Coefficient (m): ', lr.coef_)

print('Linear Model Coefficient (b): ',
lr.intercept_)

"""## 5) EVALUATE TRAINED MODEL PERFORMANCE"""

y_pred

```

```

# Plotting the TRAIN DATA

plt.figure(figsize=(16,8))

plt.scatter(X_train, y_train, color = 'gray')

plt.plot(X_train, lr.predict(X_train), color =
'red')

plt.ylabel('Marks')

plt.xlabel('Number of Hours Studied')

plt.title('Marks vs Hours Studied')


"""## 6) Making Predictions

- Use the trained model to obtain the salary
corresponding to employees who have 5 years of
experience

"""

new_value = [[5.0]]

new_prediction = lr.predict(new_value)

new_prediction

```

```

"""## 7) Save the Model"""

import joblib

model_file =
open("linear_regression_student_marks.pkl", "wb")

joblib.dump(lr, model_file)

model_file.close()

files.download('linear_regression_student_marks.pkl
')
```

2. Kode Program regression.py:

```

# Nama      : Muhammad Efflin Rizqallah Limbong

# Absen   : 225371440007
```

```

# Core Pkgs

import streamlit as st

import sklearn

import joblib,os

import numpy as np


# Loading Models

def load_prediction_model(model_file):

    loaded_model = joblib.load(open(os.path.join(model_file), "rb"))

    return loaded_model


def main():

    """Regresi Linier Sederhana"""

    st.title("Prediksi Nilai Siswa")

```

```

html_templ = """

<div
style="background-color:#726E6D;padding:10px;">

    <h3 style="color:white">Prediksi Nilai Siswa
Menggunakan Regresi Linier</h3>

</div>

"""

st.markdown(html_templ,unsafe_allow_html=True)

activity = ["Prediksi Nilai Siswa","Apa itu
Regresi?"]

choice = st.sidebar.selectbox("Menu",activity)

# Salary Determination CHOICE

if choice == 'Prediksi Nilai Siswa':

    st.subheader("Prediksi Nilai Siswa")

```

```

        jamBelajar = st.slider("Berapa Jam Siswa
tersebut belajar?",0,24)

        #st.write(type(jamBelajar))

        if st.button("Proses"):

            regressor =
load_prediction_model("linear_regression_student_ma
rks.pkl")

            jamBelajar_reshaped =
np.array(jamBelajar).reshape(-1,1)

            nilai =
regressor.predict(jamBelajar_reshaped)

            nilai = nilai[0][0] if nilai[0][0]
<= 100 else float(100)

            st.info("Nilai siswa yang belajar
selama {} jam: {:.2f}".format(jamBelajar,(nilai)))

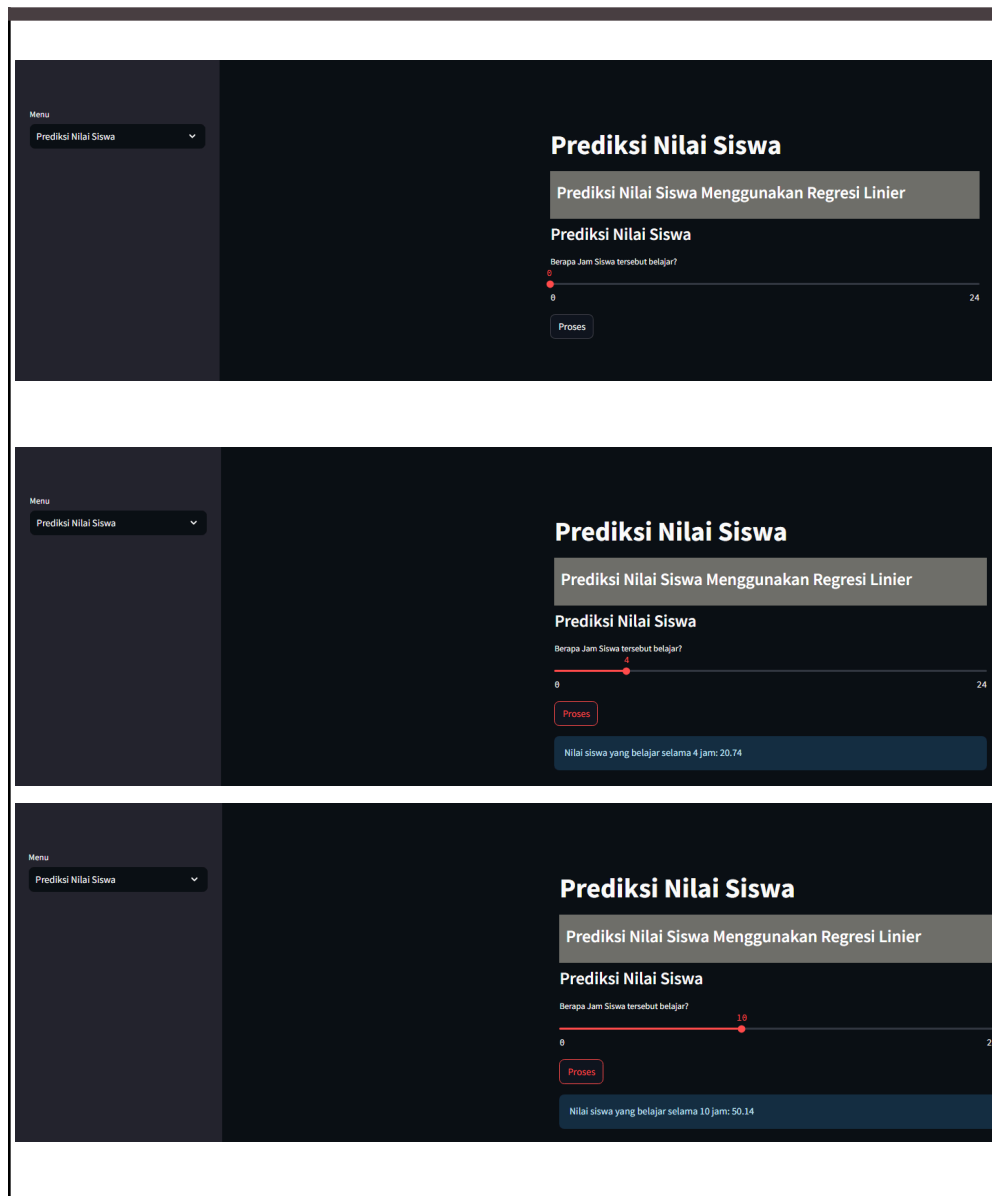
if __name__ == '__main__':

```



```
main()
```

3. Screenshot:



C. Kendala yang Dialami

Untuk praktikum ini, saya mengalami kendala dalam memahami kode pada file regression.py. Saya perlu penjelasan lebih dalam dan yang lebih mudah untuk dipahami.

D. Kesimpulan

Pada praktikum ini, kami berhasil melakukan eksplorasi data dan melatih model regresi linier sederhana menggunakan dataset nilai siswa dan jam belajar. Prosesnya melibatkan beberapa langkah utama, termasuk:

- 1.) Eksplorasi Data: Dilakukan analisis eksploratif untuk memahami karakteristik data. Kami membersihkan data dari nilai-nilai negatif, menghitung statistik deskriptif, dan memvisualisasikan hubungan antara variabel menggunakan plot seperti histogram, heatmap, dan regresi linier.
- 2.) Pembagian Data: Dataset dibagi menjadi dua bagian, yaitu data latih (training) dan data uji (testing), menggunakan fungsi `train_test_split` dari scikit-learn. Pembagian data dengan rasio 75:25 digunakan untuk menjaga kualitas evaluasi model.
- 3.) Pelatihan Model: Model regresi linier dilatih menggunakan data latih, dan kami mendapatkan parameter model, yaitu koefisien (slope) dan intercept. Koefisien ini menunjukkan pengaruh jumlah jam belajar terhadap nilai siswa.
- 4.) Evaluasi Model: Model diuji dengan data uji, dan kami menghitung kesalahan kuadrat rata-rata (Mean Squared Error - MSE) untuk mengevaluasi performa model. Kami juga memvisualisasikan model terhadap data latih untuk melihat kecocokan garis regresi dengan data asli.
- 5.) Prediksi: Model digunakan untuk melakukan prediksi nilai siswa berdasarkan jumlah jam belajar tertentu. Kami juga membangun aplikasi Streamlit sederhana yang memungkinkan pengguna memasukkan jumlah jam belajar dan melihat prediksi nilai siswa secara interaktif.

Secara keseluruhan, model regresi linier sederhana ini menunjukkan hubungan yang cukup baik antara jam belajar dan nilai siswa, serta memberikan prediksi yang memadai. Namun, akurasi model bisa ditingkatkan dengan menggunakan lebih banyak data atau mempertimbangkan faktor-faktor lain yang mungkin mempengaruhi performa akademik siswa.