

PENGEMBANGAN SISTEM SEMI OTOMATISASI UNTUK PENGATURAN ATAP JAKARTA INTERNATIONAL STADIUM DENGAN METODE KLASIFIKASI HUJAN

Dimitri Aulia Rasyidin
S1 Data Sains
Fakultas Informatika
Telkom University
dimitriauliarasyidin28@gmail.com

Dirga Patanduk
S1 Data Sains
Fakultas Informatika
Telkom University
dirgapatanduk@gmail.com

Chintya Annisah Solin
S1 Data Sains
Fakultas Informatika
Telkom University
Chintyaanisa10@gmail.com

Abstrak— Penelitian ini bertujuan mengembangkan sistem deteksi hujan menggunakan metode klasifikasi CatBoost untuk pengaturan atap otomatis di Jakarta International Stadium (JIS) dengan memanfaatkan dataset cuaca dari Kaggle yang mencakup parameter seperti suhu, kelembaban, curah hujan, durasi sinar matahari, kecepatan angin, dan arah angin. Tujuan utama penelitian ini adalah menganalisis akurasi prediksi cuaca di sekitar JIS, mengembangkan prototipe sistem dengan output 'YA' atau 'TIDAK' terkait prediksi cuaca, serta mengintegrasikan hasil 'YA' untuk mengotomatisasi penutupan atap stadion. Penelitian ini juga bertujuan untuk menerapkan ilmu sains data pada JIS dan menjadi pionir dalam penggunaan teknologi pembelajaran mesin di bidang arsitektur. Dengan algoritma CatBoost, diharapkan model prediksi cuaca yang akurat dan andal dapat dihasilkan, mendukung pengaturan atap otomatis di JIS, serta memberikan kontribusi signifikan dalam penerapan teknologi pembelajaran mesin pada sistem arsitektur modern..

Kata Kunci— Catboost, Hujan, Klasifikasi

I. Pendahuluan

Penggunaan teknologi dalam arsitektur modern semakin berkembang, salah satunya adalah penerapan sistem otomatisasi dalam pengaturan infrastruktur stadion. Jakarta International Stadium (JIS) merupakan salah satu stadion dengan fasilitas atap yang dapat dibuka dan ditutup secara otomatis. Namun, untuk mengoptimalkan sistem ini diperlukan prediksi cuaca yang akurat agar atap dapat ditutup saat hujan dan dibuka kembali saat cuaca cerah. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem

deteksi hujan yang memanfaatkan metode pembelajaran mesin, khususnya algoritma CatBoost.

CatBoost adalah algoritma boosting berbasis pohon yang dirancang untuk menangani dataset dengan fitur kategorikal secara efektif dan efisien. Algoritma ini telah terbukti unggul dalam berbagai tugas klasifikasi dan regresi, termasuk dalam prediksi multi fitur di berbagai domain seperti keuangan, kesehatan, dan meteorologi [1]. Keunggulan CatBoost dalam menangani data yang heterogen dan multivariate membuatnya menjadi pilihan yang tepat untuk prediksi cuaca yang melibatkan banyak variabel cuaca seperti suhu, kelembaban, curah hujan, dan kecepatan angin [2].

Penelitian ini mengambil dataset dari Kaggle yang mencakup berbagai parameter cuaca seperti suhu minimum, suhu maksimum, suhu rata-rata, kelembaban rata-rata, curah hujan, durasi sinar matahari, kecepatan angin maksimum, arah angin pada kecepatan maksimum, kecepatan angin rata-rata, dan arah angin dominan. Dataset ini juga mencakup indikator terjadinya banjir sebagai salah satu variabel yang dapat mempengaruhi akurasi prediksi cuaca [3].

Rumusan masalah yang menjadi fokus dalam penelitian ini adalah: (1) Bagaimana cara menganalisis akurasi prediksi cuaca di sekitar JIS? (2) Bagaimana cara mengembangkan prototipe sistem yang memberikan output 'YA' atau 'TIDAK' terkait prediksi cuaca hujan? (3) Bagaimana cara mengintegrasikan hasil prediksi 'YA'

untuk semi mengotomatisasi penutupan atap stadion? Berdasarkan rumusan masalah ini, penelitian ini memiliki tujuan untuk: (1) Menganalisis akurasi prediksi cuaca di sekitar JIS, (2) Mengembangkan prototipe sistem yang memberikan output 'YA' atau 'TIDAK' terkait prediksi cuaca, (3) Mengintegrasikan hasil 'YA' untuk semi mengotomatisasi penutupan atap, (4) Mengimplementasikan ilmu sains data pada JIS, dan (5) Menjadi pionir dalam kemajuan teknologi pembelajaran mesin pada ranah arsitektur.

Dengan menggunakan CatBoost, penelitian ini diharapkan dapat menghasilkan model prediksi cuaca yang akurat dan andal. Hasil penelitian ini akan mendukung pengaturan atap otomatis di JIS, serta memberikan kontribusi signifikan dalam penerapan teknologi pembelajaran mesin pada sistem arsitektur modern.

II. Dasar Teori

A. Kaggle

Kaggle merupakan platform online yang menyediakan berbagai dataset dari berbagai bidang, termasuk meteorologi, keuangan, kesehatan, dan lain-lain. Platform ini dikenal sebagai sumber data yang kaya dan terpercaya, sering digunakan oleh peneliti dan praktisi data untuk melatih, menguji, dan membandingkan model pembelajaran mesin. Dataset yang digunakan dalam penelitian ini diambil dari Kaggle, yang berjudul "Climate and Flood Jakarta" oleh Christopher Richard [1]. Dataset ini mencakup berbagai parameter cuaca seperti suhu, kelembaban, curah hujan, durasi sinar matahari, kecepatan angin, dan arah angin, serta data terkait banjir di Jakarta. Dengan menggunakan dataset ini, peneliti dapat mengembangkan model prediksi cuaca yang akurat untuk pengaturan atap otomatis di Jakarta International Stadium (JIS).

B. Prediksi Hujan

Prediksi hujan merupakan salah satu aplikasi penting dalam bidang meteorologi yang bertujuan untuk memperkirakan

terjadinya hujan berdasarkan analisis data cuaca historis. Prediksi yang akurat dapat membantu dalam pengambilan keputusan di berbagai sektor, termasuk pertanian, transportasi, dan manajemen infrastruktur. Dalam konteks penelitian ini, prediksi hujan sangat penting untuk mengoptimalkan pengaturan atap otomatis di JIS. Sistem ini diharapkan dapat memberikan notifikasi 'YA' atau 'TIDAK' terkait kemungkinan hujan, sehingga memungkinkan penutupan atap secara semi otomatis untuk melindungi fasilitas stadion dan kenyamanan penonton. Dengan demikian, penelitian ini tidak hanya bertujuan untuk mengembangkan model prediksi yang akurat tetapi juga untuk mengimplementasikan hasil prediksi dalam sistem otomatisasi yang nyata.

C. Metode CatBoost

CatBoost (Categorical Boosting) adalah algoritma pembelajaran mesin berbasis gradient boosting yang dikembangkan oleh Yandex. Algoritma ini dirancang untuk menangani data kategorikal dengan lebih efektif dibandingkan algoritma boosting tradisional. CatBoost menggunakan metode khusus untuk menangani variabel kategorikal tanpa perlu pra-pemrosesan yang ekstensif, sehingga dapat meningkatkan akurasi dan kecepatan pelatihan model.

Keunggulan utama CatBoost adalah kemampuannya untuk mengurangi overfitting dan bias prediktif, yang sangat penting dalam tugas-tugas klasifikasi yang kompleks seperti prediksi cuaca. Algoritma ini juga memiliki fitur yang membuatnya robust terhadap data yang hilang dan variabel dengan distribusi yang tidak merata. Dalam penelitian ini, CatBoost digunakan untuk membangun model prediksi hujan berdasarkan berbagai parameter cuaca yang ada dalam dataset. Pemilihan CatBoost didasarkan pada performanya yang unggul dalam menangani data *multivariate* dan heterogen, yang sesuai dengan karakteristik data cuaca yang kompleks dan beragam.

III. Tahapan Penelitian

A. Penambahan Data

Penelitian ini dimulai dengan tahap penambahan data, yang mencakup pengumpulan, pemrosesan, dan persiapan data untuk analisis lebih lanjut. Data yang digunakan dalam penelitian ini diambil dari Kaggle dengan judul "Climate and Flood Jakarta" oleh Christopher Richard. Dataset ini mencakup berbagai parameter cuaca yang relevan untuk prediksi hujan di wilayah Jakarta.

```
# Baca data
df = pd.read_csv('data_finish_labelled.csv')
df.tail()
```

B. Inisiasi Pustaka

Untuk mendukung implementasi penelitian ini, berbagai pustaka (*library*) Python digunakan untuk memfasilitasi penambahan data, pemodelan, dan evaluasi model. Pustaka yang digunakan mencakup pustaka untuk pengolahan data, pembelajaran mesin, dan visualisasi data. Berikut adalah pustaka-pustaka yang digunakan beserta fungsinya dalam penelitian ini:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.ensemble import GradientBoostingClassifier
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from catboost import CatBoostClassifier
```

C. Eksplorasi Data

Eksplorasi data merupakan langkah penting dalam tahap penelitian untuk memahami karakteristik dataset yang digunakan. Pada tahap ini, berbagai informasi statistik dan visualisasi diperoleh untuk mendapatkan wawasan yang lebih dalam terkait dengan data cuaca yang digunakan dalam penelitian ini.

date	Tx	Tavg	RH_avg	RR	ss	ff_x	ddd_x	ff_avg	ddd_car	station_id	station_name	region_name	flood	label_hujan		
3128	12210200	24.8	29.6	27.4	78.0	4.2	2.4	5.0	296.0	2.0	C	90745	Stasiun Meteorologi Kemayoran	Jakarta Pusat	0	YES
3132	12030000	25.6	34.8	29.6	69.0	0.0	2.0	7.0	260.0	2.0	W	90745	Stasiun Meteorologi Kemayoran	Jakarta Pusat	0	NO
3133	12040000	26.4	33.4	28.9	68.0	0.0	7.0	5.0	300.0	2.0	C	90745	Stasiun Meteorologi Kemayoran	Jakarta Pusat	0	NO
3134	12090000	26.4	32.0	26.3	73.0	0.0	6.3	6.0	296.0	2.0	C	90745	Stasiun Meteorologi Kemayoran	Jakarta Pusat	0	NO
3135	12101000	25.6	30.2	26.5	83.0	5.0	0.2	4.0	240.0	1.0	C	90745	Stasiun Meteorologi Kemayoran	Jakarta Pusat	0	YES

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3136 entries, 0 to 3135
Data columns (total 16 columns):
#   Column              Non-Null Count  Dtype
---  -
0   date                 3136 non-null   object
1   Tn                   3136 non-null   float64
2   Tx                   3136 non-null   float64
3   Tavg                 3136 non-null   float64
4   RH_avg               3136 non-null   float64
5   RR                   3136 non-null   float64
6   ss                   3136 non-null   float64
7   ff_x                 3136 non-null   float64
8   ddd_x                3136 non-null   float64
9   ff_avg               3136 non-null   float64
10  ddd_car              3136 non-null   object
11  station_id           3136 non-null   int64
12  station_name         3136 non-null   object
13  region_name          3136 non-null   object
14  flood                3136 non-null   int64
15  label_hujan          3136 non-null   object
dtypes: float64(9), int64(2), object(5)
memory usage: 392.1+ KB
```

```
1 unique_values = df['ddd_car'].unique()
2
3 # Menampilkan nilai unik
4 print("Nilai unik di kolom 'ddd_car':")
5 for value in unique_values:
6     print(value)
```

Nilai unik di kolom 'ddd_car':

W
SW
E
S
NE
SE
C
N
NW

kolom 'ddd_car' yang mengandung arah angin dalam format string perlu diubah menjadi tipe data numerik. Transformasi ini diperlukan agar kolom tersebut dapat dimasukkan ke dalam model pembelajaran mesin dengan benar. Misalnya, jika arah angin direpresentasikan dengan kata-kata seperti 'W' untuk Barat, 'SW' untuk Barat Daya, dan seterusnya, perlu dilakukan pemetaan ke nilai numerik yang sesuai.

Eksplorasi data juga melibatkan visualisasi seperti histogram untuk distribusi nilai dalam setiap fitur, serta plot yang membandingkan relasi antara variabel yang berbeda. Tujuannya adalah untuk menemukan pola, outlier, atau keanehan lainnya yang mungkin mempengaruhi kualitas model prediksi. Untuk Eksplorasi

data kami menggunakan pustaka PandasProfiling

```
from ydata_profiling import ProfileReport

profiling = ProfileReport(df, title="Jakarta Rainfall")
profiling.to_notebook_iframe()
```

Langkah-langkah ini penting untuk memastikan bahwa data dipahami secara mendalam sebelum dilakukan pemodelan, sehingga hasil analisis dan prediksi yang dihasilkan lebih reliabel dan akurat.

D. Transformasi Data

Transformasi fitur dari tipe data string menjadi numerik diperlukan untuk memungkinkan penggunaan fitur-fitur tersebut dalam pembangunan model pembelajaran mesin. Dalam penelitian ini, fitur yang akan diubah adalah kolom 'ddd_car', yang merepresentasikan arah angin dalam bentuk string.

```
# Dictionary untuk memetakan nilai-nilai lama ke nilai-nilai baru
mapping = {
    'W': 1,
    'SW': 2,
    'E': 3,
    'S': 4,
    'NE': 5,
    'SE': 6,
    'C': 7,
    'N': 8,
    'NW': 9
}

# Kolom ddd_car yang ingin diubah
kolom_ddd_car = ['W', 'SW', 'E', 'S', 'NE', 'SE', 'C', 'N', 'NW']

# Loop melalui kolom ddd_car dan ubah nilai sesuai dengan mapping
for nilai_lama in kolom_ddd_car:
    # Gunakan nilai_lama sebagai kunci untuk mendapatkan nilai baru dari dictionary mapping
    nilai_baru = mapping[nilai_lama]
    # Ubah nilai lama dengan nilai baru di DataFrame
    df.loc[df['ddd_car'] == nilai_lama, 'ddd_car'] = nilai_baru

# Melakukan casting kolom 'ddd_car' dari object menjadi int
df['ddd_car'] = df['ddd_car'].astype(int)

# Menggunakan metode replace() untuk mengubah nilai YES menjadi 1 dan NO menjadi 0
df['label_hujan'] = df['label_hujan'].replace({'YES': 1, 'NO': 0})

# Menampilkan DataFrame setelah perubahan
df.head()
```

Setelah transformasi ini, DataFrame df siap digunakan untuk tahap selanjutnya dalam penelitian, seperti pembagian dataset untuk pelatihan dan pengujian model, serta pembangunan model prediksi menggunakan algoritma CatBoost. Transformasi ini memastikan bahwa semua fitur yang relevan dalam dataset telah disiapkan dengan benar untuk analisis lebih lanjut.

E. Kondisi Data Pasca Pra Proses

Setelah tahap pra-pemrosesan data, penting untuk mengevaluasi kondisi data untuk memastikan bahwa semua transformasi dan

persiapan telah dilakukan dengan benar. Berikut adalah hasil evaluasi kondisi data pasca pra-proses menggunakan fungsi df.info():

	date	Tn	Tx	Tavg	RH_avg	RR	ss	ff_x	ddd_x	ff_avg	ddd_car	station_id	station_name	region_name	flood	label_hujan
0	10/20/16	25.6	33.2	27.0	86.0	1.6	6.7	4.0	200.0	2.0	1	96733	Stasiun Klimatologi Baran	Jakarta Selatan	1	1
1	10/20/16	24.4	34.9	26.1	86.0	3.0	5.4	4.0	200.0	2.0	2	96733	Stasiun Klimatologi Baran	Jakarta Selatan	1	1
2	10/20/16	25.0	33.6	28.9	86.0	1.6	5.6	4.0	199.0	2.0	3	96733	Stasiun Klimatologi Baran	Jakarta Selatan	0	1
3	10/20/16	26.2	31.2	28.0	82.0	11.5	6.7	3.0	60.0	1.0	4	96733	Stasiun Klimatologi Baran	Jakarta Selatan	1	1
4	10/20/16	24.8	33.2	26.3	92.0	36.3	6.9	3.0	270.0	1.0	3	96733	Stasiun Klimatologi Baran	Jakarta Selatan	0	1

Dengan evaluasi ini, peneliti dapat memastikan bahwa semua fitur yang diperlukan telah siap digunakan untuk analisis lebih lanjut, seperti pembagian dataset, pembangunan model prediksi, dan evaluasi performa model. Evaluasi kondisi data yang teliti setelah pra-proses adalah langkah kritis dalam memastikan kualitas dan integritas data sepanjang tahap penelitian.

1 df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3136 entries, 0 to 3135
Data columns (total 16 columns):
#   Column          Non-Null Count  Dtype
---  -
0   date             3136 non-null   object
1   Tn               3136 non-null   float64
2   Tx              3136 non-null   float64
3   Tavg            3136 non-null   float64
4   RH_avg          3136 non-null   float64
5   RR              3136 non-null   float64
6   ss              3136 non-null   float64
7   ff_x            3136 non-null   float64
8   ddd_x           3136 non-null   float64
9   ff_avg          3136 non-null   float64
10  ddd_car          3136 non-null   int32
11  station_id       3136 non-null   int64
12  station_name     3136 non-null   object
13  region_name      3136 non-null   object
14  flood            3136 non-null   int64
15  label_hujan      3136 non-null   int64
dtypes: float64(9), int32(1), int64(3), object(3)
memory usage: 379.9+ KB
```

F. Pembagian Kolom Fitur dan Label

Pada tahap ini, dataset yang telah diproses akan dibagi menjadi dua bagian utama: fitur (X) dan label (y). Proses ini diperlukan untuk melatih model prediksi dan menguji performanya menggunakan data yang terpisah. Berikut adalah implementasi pembagian kolom fitur dan label dari dataset:

```
# Pisahkan fitur dan target
X = df[['Tn', 'Tx', 'ff_avg', 'ff_x', 'Tavg', 'ddd_car', 'ss', 'ddd_x', 'RH_avg']]
y = df['label_hujan']
```

Pembagian ini penting karena memungkinkan model pembelajaran mesin

untuk mempelajari pola dari fitur-fitur tertentu dalam data pelatihan (X), dan kemudian melakukan prediksi berdasarkan fitur-fitur tersebut untuk label (y). Setelah pembagian ini, dataset siap untuk langkah-langkah selanjutnya dalam pembangunan model, seperti pembagian data menjadi set pelatihan dan pengujian, serta pelatihan dan evaluasi model prediksi.

G. Pembagian data Latih dan Uji

Pembagian data menjadi data latih dan data uji adalah langkah krusial dalam pembangunan model prediksi. Hal ini dilakukan untuk menguji performa model pada data yang tidak terlihat selama pelatihan. Berikut adalah implementasi pembagian data latih dan uji menggunakan fungsi `train_test_split` dari pustaka Scikit-learn:

```
# Bagi data menjadi data Latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.25,
                                                    random_state=42)
```

Pembagian ini melibatkan 75% isi data untuk menjadi data latih dan 25% dari data untuk menjadi data uji.

H. Pemodelan

Pada tahap pemodelan ini, menggunakan algoritma CatBoost untuk klasifikasi, kami melakukan eksperimen dengan berbagai jumlah iterasi untuk mengoptimalkan akurasi prediksi. Proses dimulai dengan inisialisasi model CatBoost dan iterasi melalui 100 langkah. Setiap iterasi melibatkan pelatihan model menggunakan data latih (X_train dan y_train) dan pengujian pada data uji (X_test dan y_test). Selama setiap iterasi, kami menghitung akurasi prediksi untuk kedua dataset tersebut menggunakan metrik akurasi.

Untuk memonitor performa model, kami menyimpan histori akurasi dari setiap iterasi dalam `train_accuracy_history` dan `test_accuracy_history`. Selain itu, kami juga melacak iterasi di mana model mencapai akurasi tertinggi pada data latih (`best_train_accuracy_cat`) dan data uji

(`best_test_accuracy_cat`). Hasil eksperimen menunjukkan bahwa akurasi tertinggi untuk data latih adalah `{best_train_accuracy_cat}` pada iterasi `{best_train_iter}`, sementara untuk data uji, akurasi tertinggi adalah `{best_test_accuracy_cat}` pada iterasi `{best_test_iter}`.

```
# List untuk menyimpan akurasi dari setiap iterasi
train_accuracy_history = []
test_accuracy_history = []

# Variabel untuk melacak iterasi dengan akurasi tertinggi
best_train_accuracy_cat = 0
best_train_iter = 0
best_test_accuracy_cat = 0
best_test_iter = 0

# Menggunakan model CatBoost yang baru untuk setiap iterasi
for i in range(1, 101): # 100 iterasi
    # Inisialisasi model CatBoost
    catboost_model = CatBoostClassifier(iterations=i, verbose=False)

    # Latih model
    catboost_model.fit(X_train, y_train)

    # Prediksi pada data latih dan data uji
    y_pred_train = catboost_model.predict(X_train)
    y_pred_test = catboost_model.predict(X_test)

    # Hitung akurasi
    train_accuracy = accuracy_score(y_train, y_pred_train)
    test_accuracy = accuracy_score(y_test, y_pred_test)

    # Simpan akurasi ke dalam list
    train_accuracy_history.append(train_accuracy)
    test_accuracy_history.append(test_accuracy)

    # Update iterasi terbaik untuk data latih
    if train_accuracy > best_train_accuracy_cat:
        best_train_accuracy_cat = train_accuracy
        best_train_iter = i

    # Update iterasi terbaik untuk data uji
    if test_accuracy > best_test_accuracy_cat:
        best_test_accuracy_cat = test_accuracy
        best_test_iter = i

# Print akurasi terbaik dari 100 iterasi
print(f'Best Train Accuracy: {best_train_accuracy_cat} at iteration {best_train_iter}')
print(f'Best Test Accuracy: {best_test_accuracy_cat} at iteration {best_test_iter}')
```

Proses ini memberikan pandangan yang jelas tentang kemajuan dan performa model selama iterasi, memungkinkan kami untuk memilih model terbaik berdasarkan akurasi yang dicapai. Evaluasi akurasi ini penting untuk memastikan bahwa model yang dikembangkan mampu memberikan prediksi yang akurat dan dapat diandalkan untuk aplikasi praktis dalam mendeteksi kemungkinan hujan di sekitar Jakarta International Stadium berdasarkan kondisi cuaca.

I. Ekstraksi Model

Setelah menemukan model CatBoost dengan akurasi terbaik pada data uji, langkah selanjutnya adalah mengekstrak dan menyimpan model tersebut untuk penggunaan di masa depan. Dalam kode berikut, kami menggunakan fungsi `save_model` dari CatBoost untuk menyimpan model ke dalam file `best_catboost_model.bin`:

```
# Simpan model CatBoost yang memiliki akurasi test terbaik
catboost_model.save_model('best_catboost_model.bin')
```


Proses ini memungkinkan kami untuk mengambil model yang telah dilatih dan menyimpannya dalam format biner yang dapat dengan mudah dimuat kembali untuk melakukan prediksi pada data baru atau untuk integrasi dengan aplikasi yang lebih luas, seperti prototipe sistem yang dikembangkan untuk Jakarta International Stadium. Menyimpan model yang memiliki performa terbaik adalah langkah penting dalam menjaga konsistensi dan kualitas prediksi dari hasil penelitian ini.

J. Pembangunan Antarmuka

Dalam tahap ini, kami mengimplementasikan antarmuka berbasis web menggunakan Streamlit untuk menampilkan dan menggunakan model prediksi hujan yang telah dikembangkan. Antarmuka ini memungkinkan pengguna untuk memasukkan parameter-parameter cuaca tertentu dan mendapatkan prediksi apakah akan terjadi hujan di sekitar Jakarta International Stadium, serta menampilkan rekomendasi terkait penutupan atap berdasarkan prediksi.

Kami memulai dengan memuat model CatBoost yang memiliki performa terbaik dari file `best_catboost_model.bin`. Fungsi `predict_rain` dipergunakan untuk melakukan prediksi berdasarkan input pengguna yang meliputi suhu minimum (Tn), suhu maksimum (Tx), kecepatan angin rata-rata (ff_avg), kecepatan angin maksimum (ff_x), suhu rata-rata (Tavg), arah angin dominan (ddd_car), durasi sinar matahari (ss), arah angin pada kecepatan maksimum (ddd_x), dan kelembaban rata-rata (RH).

Antarmuka dibangun dengan judul "Jakarta International Stadium Rain Detector" sebagai header utama. Input-parameter disediakan di sidebar untuk memasukkan nilai-nilai yang diperlukan untuk prediksi. Setelah pengguna memasukkan nilai dan menekan tombol "Predict", sistem akan menampilkan video yang menunjukkan rekomendasi penutupan atau pembukaan

atap berdasarkan hasil prediksi model. Jika prediksi menunjukkan kemungkinan hujan, video menunjukkan atap tertutup; sebaliknya, jika tidak ada prediksi hujan, video menunjukkan atap terbuka.

```
import streamlit as st
import numpy as np
from catboost import CatBoostClassifier

# Load model
best_catboost_model = CatBoostClassifier()
best_catboost_model.load_model('best_catboost_model.bin')

# Mapping for wind direction
ddd_car_mapping = {'N': 1, 'NE': 2, 'E': 3, 'SE': 4, 'S': 5, 'SW': 6, 'W': 7, 'NW': 8}

# Function to predict rain
def predict_rain(Tn, Tx, ff_avg, ff_x, Tavg, ddd_car, ss, ddd_x, RH):
    # Convert wind direction to numeric value
    ddd_car_numeric = ddd_car_mapping.get(ddd_car, 0)

    # Prepare input features
    input_features = np.array([[Tn, Tx, ff_avg, ff_x, Tavg, ss, ddd_x, ddd_car_numeric, RH]])

    # Predict using model
    prediction = best_catboost_model.predict(input_features)

    return prediction

st.header("Jakarta International Stadium Rain Detector", anchor=None, divider=False)
# st.image("https://static-2.tftt.net/tr/turnnews/16/Foto/barek/images/Jakarta-International-Stadium-112333.jpg", caption=None, width=600, use_column_width=None, clamp=False, channels="RGB", output_format="auto")
# Input form
st.sidebar.title("Input Parameters")
Tn = st.sidebar.number_input("Masukkan nilai min temperature (°C) Tn: ", min_value=22.0, max_value=50.0, value=22.0)
Tx = st.sidebar.number_input("Masukkan nilai max temperature (°C) Tx: ", min_value=22.0, max_value=50.0, value=22.0)
ff_avg = st.sidebar.number_input("Masukkan nilai average wind speed (m/s) ff_avg: ", min_value=0.0, max_value=100.0, value=50.0)
ff_x = st.sidebar.number_input("Masukkan nilai max wind speed (m/s) ff_x: ", min_value=0.0, max_value=100.0, value=15.0)
Tavg = st.sidebar.number_input("Masukkan nilai avg temperature (°C) Tavg: ", min_value=22.0, max_value=50.0, value=22.0)
ddd_car = st.sidebar.selectbox("Masukkan wind direction (*) ddd_car: ", list(ddd_car_mapping.keys()))
ss = st.sidebar.number_input("Masukkan nilai duration of sunshine (hour) ss: ", min_value=0.0, max_value=24.0, value=0.0)
ddd_x = st.sidebar.number_input("Masukkan nilai wind direction at maximum speed (*) ddd_x: ", min_value=0, max_value=100, value=100)
RH = st.sidebar.number_input("Masukkan nilai avg humidity(S) RH: ", min_value=0, max_value=100, value=55)

# Predict
if st.sidebar.button("Predict"):
    prediction = predict_rain(Tn, Tx, ff_avg, ff_x, Tavg, ddd_car, ss, ddd_x, RH)
    if prediction[0] == 1:
        st.video("close_jis.mp4", format="video/mp4", start_time=0, autoplay=True)
        st.write("Berdasarkan input yang diberikan, model memprediksi bahwa kemungkinan akan ada hujan.\n Atap ditutup.")
    else:
        st.video("open_jis.mp4", format="video/mp4", start_time=0, autoplay=True)
        st.write("Berdasarkan input yang diberikan, model memprediksi bahwa kemungkinan tidak akan ada hujan.\n Atap dibuka.")
```

Antarmuka ini tidak hanya memperlihatkan integrasi langsung antara model machine learning dan aplikasi praktis, tetapi juga memberikan contoh konkret penerapan ilmu data dalam konteks pengaturan stadion untuk meningkatkan pengalaman pengguna.

IV. Hasil dan Pembahasan

A. Hasil Akurasi Model

Pada tahap pemodelan menggunakan CatBoost untuk prediksi cuaca di sekitar Jakarta International Stadium, kami mencapai hasil akurasi yang signifikan. Hasil terbaik yang diperoleh dari eksperimen ini adalah akurasi pada data latih sebesar 87.20% pada iterasi ke-48 dan akurasi pada data uji sebesar 84.69% pada iterasi ke-2. Akurasi ini mencerminkan kemampuan model dalam mengklasifikasikan kondisi cuaca berdasarkan sejumlah variabel seperti suhu, kecepatan angin, durasi sinar matahari, arah angin, dan kelembaban.

Akurasi yang tinggi pada data latih menunjukkan bahwa model mampu memahami pola dan hubungan antar variabel dalam dataset pelatihan dengan baik. Sementara itu, meskipun terjadi sedikit penurunan dalam akurasi pada data uji dibandingkan dengan data latih, hasil ini masih menunjukkan bahwa model memiliki kemampuan umum yang baik dalam menggeneralisasi pola-pola yang dipelajari ke data baru yang belum pernah dilihat sebelumnya.

Pencapaian akurasi yang solid ini mendukung validitas dan keandalan model CatBoost dalam aplikasi prediksi cuaca, khususnya dalam konteks pengaturan penutupan atap stadion berdasarkan prediksi hujan. Meskipun ada variasi dalam akurasi antara data latih dan uji, hasil tersebut konsisten dengan karakteristik umum dari proses pembelajaran mesin di mana tujuan utama adalah untuk menghasilkan model yang dapat bekerja dengan baik pada data baru yang belum pernah dilihat sebelumnya.

```
Best Train Accuracy: 0.8720238095238095 at iteration 48  
Best Test Accuracy: 0.8469387755102041 at iteration 2
```

Hasil akurasi yang didapatkan menunjukkan bahwa model CatBoost yang digunakan dalam penelitian ini berhasil dalam memprediksi kemungkinan hujan di sekitar Jakarta International Stadium berdasarkan kondisi cuaca yang diberikan. Dengan akurasi pada tingkat yang tinggi seperti ini, implementasi praktis dari model dalam pengaturan penutupan atap stadion dapat memberikan manfaat yang signifikan, termasuk dalam mengoptimalkan pengalaman pengguna dan pengelolaan acara olahraga.

Namun demikian, ada beberapa pertimbangan yang perlu diperhatikan terkait hasil ini. Variabilitas antara akurasi pada data latih dan data uji mengindikasikan adanya potensi untuk meningkatkan generalisasi model terhadap data baru yang lebih beragam. Untuk

penelitian lebih lanjut, pemilihan fitur yang lebih tepat dan penambahan data cuaca yang lebih khusus untuk wilayah Jakarta dapat meningkatkan performa model, terutama dalam menghadapi variasi musiman dan faktor cuaca lokal yang lebih kompleks.

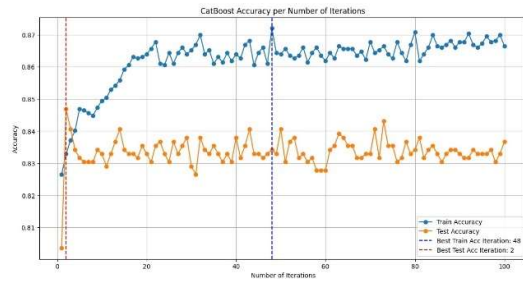
Selain itu, evaluasi lebih lanjut terhadap performa model dalam konteks aplikasi praktis, seperti pengaruh dari keputusan penutupan atap terhadap pengalaman penonton dan kondisi lapangan, dapat menjadi fokus untuk memastikan bahwa penggunaan model ini tidak hanya akurat tetapi juga menghasilkan dampak positif secara praktis.

B. Visualisasi Hasil Pelatihan Model

Visualisasi hasil pelatihan model CatBoost menunjukkan pola yang menarik dalam proses pembelajaran mesin. Dari total 100 iterasi yang dilakukan, model mulai mencapai akurasi yang baik sejak awal, tepatnya sejak iterasi kedua. Grafik akurasi pada data latih dan data uji menunjukkan tren kenaikan yang konsisten hingga mencapai puncaknya pada iterasi ke-48.

Pada awalnya, terlihat adanya peningkatan yang cepat dalam akurasi pada data uji seiring dengan peningkatan iterasi. Namun, setelah mencapai iterasi ke-48, terjadi stagnasi dalam peningkatan performa model, di mana akurasi tidak mengalami perubahan yang signifikan bahkan cenderung datar atau turun sedikit.

Fenomena ini mengindikasikan bahwa mesin belajar (machine learning) dalam konteks ini telah mencapai titik optimal dalam pembelajaran pola-pola yang ada dalam data latih. Hal ini menandakan bahwa lebih lanjutnya iterasi tidak lagi memberikan peningkatan yang substansial dalam kemampuan model untuk menggeneralisasi ke data uji yang belum pernah dilihat sebelumnya.



Visualisasi ini penting untuk memahami bagaimana proses pembelajaran mesin berlangsung dalam konteks pengembangan model prediksi cuaca. Puncak akurasi pada iterasi ke-48 menunjukkan titik di mana model mencapai kematangan dalam kemampuannya untuk memahami dan mengklasifikasikan pola-pola cuaca yang relevan untuk prediksi hujan di sekitar Jakarta International Stadium. Evaluasi terhadap grafik ini memberikan wawasan yang berharga untuk pengoptimalan lebih lanjut terhadap model, seperti penambahan fitur atau tuning parameter, guna meningkatkan performa dan generalisasi model di masa depan.

C. Hasil Antarmuka

Antarmuka yang dikembangkan menggunakan Streamlit, seperti yang diimplementasikan berdasarkan kode yang disediakan sebelumnya, telah menghasilkan sebuah aplikasi praktis untuk prediksi cuaca di sekitar Jakarta International Stadium. Antarmuka ini memungkinkan pengguna untuk memasukkan beberapa parameter cuaca seperti suhu, kecepatan angin, durasi sinar matahari, arah angin, dan kelembaban. Setelah memasukkan nilai-nilai ini, pengguna dapat mengklik tombol "Predict" untuk melihat hasil prediksi dari model CatBoost yang telah dilatih sebelumnya.

Salah satu fitur utama dari antarmuka ini adalah kemampuannya untuk memberikan ilustrasi visual dengan video yang menampilkan simulasi atap stadion yang terbuka atau tertutup berdasarkan hasil klasifikasi dari input pengguna. Misalnya, jika model memprediksi kemungkinan

hujan, video akan menunjukkan simulasi atap tertutup untuk memberikan perlindungan terhadap cuaca buruk. Sebaliknya, jika model memprediksi tidak ada hujan, video akan menampilkan simulasi atap stadion yang terbuka.

Antarmuka juga telah dirancang dengan memperhitungkan batasan-batasan numerik yang realistis berdasarkan sejarah cuaca di Jakarta, memastikan bahwa input dari pengguna sesuai dengan rentang yang relevan dan dapat dipercaya untuk prediksi yang akurat. Desain antarmuka yang sederhana dan mudah dipahami oleh pengguna juga menjadi nilai tambah, memungkinkan pengguna dari berbagai latar belakang untuk menggunakan aplikasi ini dengan nyaman tanpa memerlukan pengetahuan teknis yang mendalam.



Keseluruhan, hasil dari implementasi antarmuka ini menunjukkan integrasi yang efektif antara model prediksi data dan aplikasi praktis dalam konteks pengelolaan cuaca di stadion. Pengembangan lebih lanjut dapat difokuskan pada peningkatan responsivitas dan interaktivitas antarmuka, serta penambahan fitur yang dapat memperkaya pengalaman pengguna dalam memanfaatkan informasi prediksi cuaca untuk pengaturan stadion yang lebih optimal.

V. Kesimpulan

Penelitian ini berhasil mengembangkan prototipe sistem deteksi hujan untuk pengaturan atap otomatis di Jakarta International Stadium (JIS) menggunakan metode klasifikasi CatBoost, yang diimplementasikan hingga aplikasi berbasis

web menggunakan Streamlit. Dengan memanfaatkan dataset cuaca dari Kaggle dan menggunakan algoritma CatBoost, sistem ini mampu memprediksi kemungkinan terjadinya hujan dengan akurasi 0.84 pada data uji dan 0.87 pada data latih. Tantangan utama dalam penelitian ini adalah sulitnya mencari dataset cuaca yang spesifik untuk klasifikasi hujan di wilayah kota Jakarta. Namun, hasil prediksi diintegrasikan ke dalam prototipe yang memberikan instruksi visual untuk membuka atau menutup atap stadion berdasarkan prediksi cuaca. Untuk penelitian lebih lanjut, disarankan agar dataset diperkaya dengan fitur tambahan dan mengadaptasi model time series guna meningkatkan akurasi dan dinamisasi prediksi, serta mendukung pengambilan keputusan yang lebih tepat dalam pengaturan atap otomatis di JIS.

VI. Referensi

- [1] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, vol. 31, pp. 6638-6648, 2018.
- [2] Y. Lou, R. Caruana, and J. Gehrke, "Intelligible models for classification and regression," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 150-158, 2012.
- [3] C. R. Christopher, "Climate and Flood Jakarta," Kaggle, [Online]. Available: <https://www.kaggle.com/datasets/christopherrichardc/climate-and-flood-jakarta>. [Accessed: February 28, 2024].