Milestone 3 Report: Model Deployment and Interactive Dashboard Development

By: Kuan-Chen, Chen

1. Project Overview

This project began with a central question: Can data science help young athletes choose the professional sport that best fits their physical attributes and potential? Inspired by real-world discussions during the Super Bowl, this idea evolved into a predictive system that compares potential athletic success across three globally popular sports leagues: the NBA (basketball), NFL (American football), and FIFA (soccer).

In Milestone 1, we gathered data from sports video games and real-world salary datasets, cleaned and merged the datasets, and conducted Exploratory Data Analysis (EDA). We focused on standardizing key variables—height, weight, age, and salary—and introduced derived features like BMI and years of professional experience.

In Milestone 2, we engineered additional features (e.g., age of entering professional leagues), encoded positions, and built multiple regression models to predict annual player salary. We evaluated models using MAE and RMSE across NBA, NFL, and FIFA datasets, and saved the best-performing models for deployment.

Milestone 3 now focuses on deploying these models into a fully interactive, real-time web dashboard using Dash. The dashboard empowers users to input their own physical data and explore potential earnings and fit across the three sports.

2. Purpose and Goals of the Dashboard

The primary goal of the dashboard is to:

Provide real-time salary predictions for users based on height, weight, age, years pro, and position.

Enable users to compare predictions across NBA, NFL, and FIFA.

Help young athletes or decision-makers better understand which sport may be the best career path.

By translating machine learning insights into an accessible tool, we aim to make predictive analytics useful in real-life decision-making scenarios, particularly for career planning in athletics.

3. Dashboard Architecture

A. Interface Design

The dashboard is built using Dash, a Python framework for creating web applications. The layout consists of an input section with sliders for height, weight, and age, a numeric input for years of professional experience, and a display for the computed BMI value. Below the input section, the application includes three sport-specific tabs—NBA, NFL, and FIFA—each containing a custom position dropdown and the corresponding salary prediction. Outputs are dynamically generated in each tab based on the selected sport and user-provided attributes.

```
app = dash.Dash(__name__, suppress_callback_exceptions=True)
app.title = "Athlete Career Prediction Dashboard"

# Define color scheme
colors = {
    'background': '#F9F9F9',
    'text': '#333333',
    'accent': '#1E88E5',
    'highlight': '#FFC107',
    'chart': '#2196F3',
    'fifa': '#EA2027',
    'nba': '#009432',
    'nfl': '#0652DD'
}

header = html.Div([
    html.H1("Athlete Career Prediction Dashboard", style={'textAlign': 'center', 'color': colors['text']}),
    html.Hr(),
    html.H4("Enter athlete data to predict overall rating changes over the next 10 years",
            style={'textAlign': 'center', 'color': colors['text']}),
    html.Hr(),
])

footer = html.Div([
    html.Hr(),
    html.H4("About This Dashboard"),
    html.P("This tool uses machine learning models to predict athletes' future overall ratings based on physical data, age, and position."),
    html.P("Note: All predictions are for reference only. Actual results may vary due to injuries, training, and other factors."),
], style={'marginTop': '30px', 'paddingTop': '20px', 'borderTop': '1px solid #eee'})

# Define app layout
app.layout = html.Div([
    header,

    # Shared metrics at the top
    create_shared_metrics(),

    # Separate panels for each sport
    create_fifa_panel(),
    create_nba_panel(),
    create_nfl_panel(),

    footer,
], style={'padding': '20px', 'backgroundColor': colors['background']})
```

B. Feature Handling and Derived Metrics

The dashboard automatically calculates Body Mass Index (BMI) based on the user's height and weight. It also determines the age at which a user hypothetically entered professional leagues (referred to as Enter_Pro), calculated by subtracting years of professional experience from the user's current age. These derived metrics are not only presented to users as reference values but are also used internally by the models, ensuring consistency between training and inference stages.

```python
def create_shared_metrics():
    return html.Div([
        html.H3("Shared Athlete Metrics", style={'marginBottom': '20px', 'textAlign': 'center'}),
        html.Div([
            html.Div([
                html.Label("Height (cm)"),
                dcc.Slider(
                    id='height-slider',
                    min=150,
                    max=230,
                    step=1,
                    value=185,
                    marks={i: str(i) for i in range(150, 231, 10)},
                    tooltip={"placement": "bottom", "always_visible": True}
                ),
                html.Label("Weight (kg)", style={'marginTop': '15px'}),
                dcc.Slider(
                    id='weight-slider',
                    min=50,
                    max=150,
                    step=1,
                    value=80,
                    marks={i: str(i) for i in range(50, 151, 10)},
                    tooltip={"placement": "bottom", "always_visible": True}
                ),
            ], className='six columns'),
            html.Div([
                html.Label("Age"),
                dcc.Slider(
                    id='age-slider',
                    min=16,
                    max=40,
                    step=1,
                    value=25,
                    marks={i: str(i) for i in range(16, 41, 4)},
                    tooltip={"placement": "bottom", "always_visible": True}
                ),
                html.Div([
                    html.Label("BMI (automatically calculated):",
                               style={'display': 'inline-block', 'marginRight': '10px', 'marginTop': '15px'}),
                    html.Div(
                        id='bmi-display',
                        style={'display': 'inline-block', 'fontWeight': 'bold', 'marginTop': '15px'}
                    )
                ]),
            ], className='six columns'),
        ], className='row'),

        html.Hr(style={'marginTop': '20px', 'marginBottom': '20px'}),
    ], style={'padding': '20px', 'backgroundColor': '#FFFFFF', 'borderRadius': '5px', 'boxShadow': '0px 0px 5px lightgrey', 'marginBottom': '20px'})
```

C. Model Integration

Three machine learning models—each corresponding to NBA, NFL, and FIFA—were trained during Milestone 2 and saved using the joblib format. These models accept structured inputs that include physical attributes (height, weight, age), derived values (BMI and Enter_Pro), and position encoding. Each time the user interacts with the dashboard by changing input values, the relevant model is triggered to compute and return the predicted salary.

```python
models = {}
try:
    models['FIFA'] = joblib.load('fifa_model.joblib')
    print("FIFA model loaded successfully!")
except Exception as e:
    print(f"FIFA model loading error: {e}")
    models['FIFA'] = None

try:
    models['NBA'] = joblib.load('nba_model.joblib')
    print("NBA model loaded successfully!")
except Exception as e:
    print(f"NBA model loading error: {e}")
    models['NBA'] = None

try:
    models['NFL'] = joblib.load('nfl_model.joblib')
    print("NFL model loaded successfully!")
except Exception as e:
    print(f"NFL model loading error: {e}")
    models['NFL'] = None
```

D. Callback Structure

Dash's callback system ensures that any change in input values automatically updates all related outputs. This includes real-time updates to the BMI display, recalculation of Enter_Pro, and regeneration of salary predictions. Callback functions are organized to reuse logic across the different sport tabs, contributing to modularity and performance.

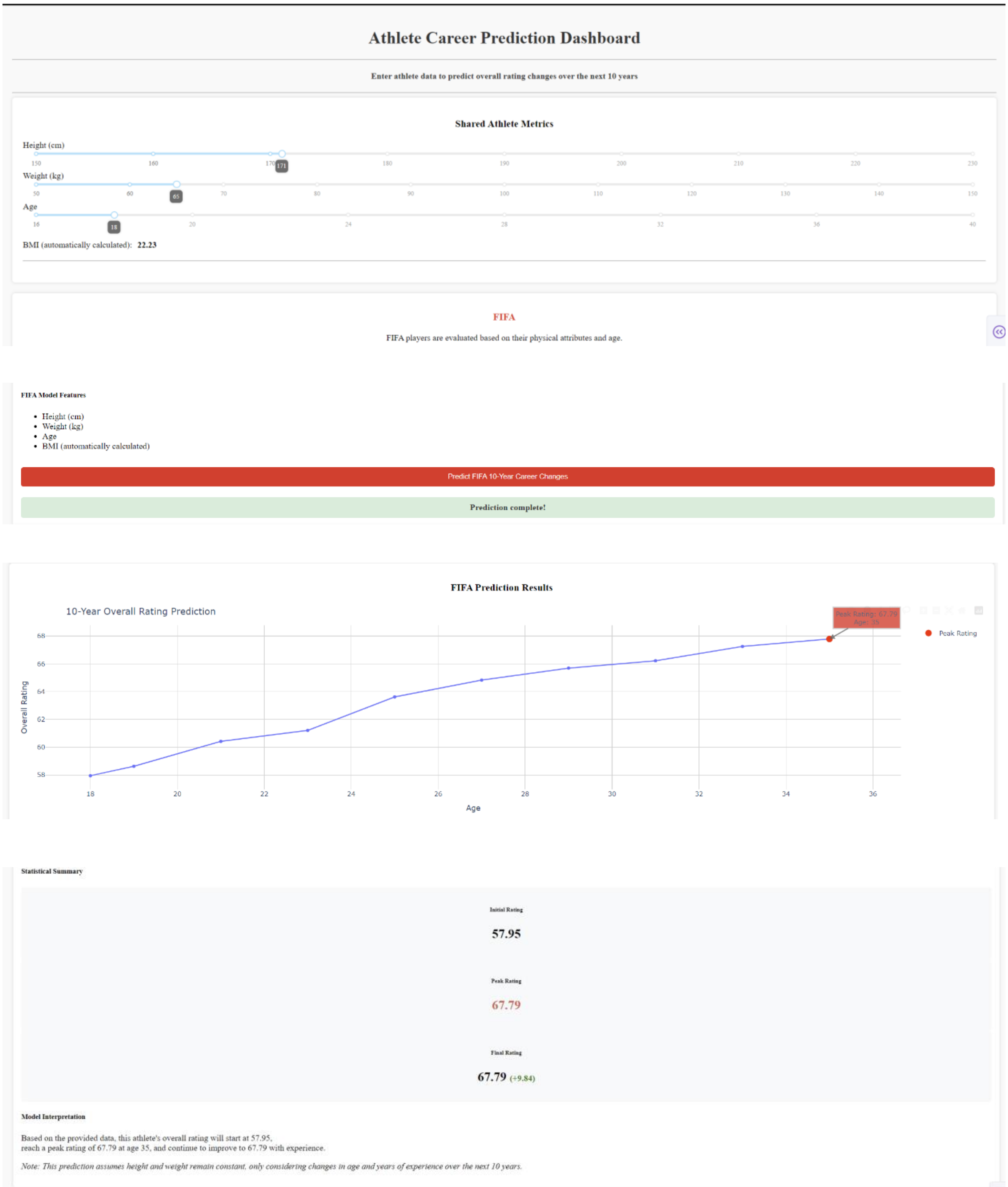
```python
@app.callback(
    Output('bmi-display', 'children'),
    [Input('height-slider', 'value'),
     Input('weight-slider', 'value')]
)
def calculate_bmi(height, weight):
    if height and weight:
        bmi = weight / ((height/100) ** 2)
        return f"{bmi:.2f}"
    return "N/A"

# Callback to automatically calculate NBA age entered pro league
@app.callback(
    Output('nba-enter-pro-display', 'children'),
    [Input('age-slider', 'value'),
     Input('nba-years-pro-slider', 'value')]
)
def calculate_nba_enter_pro(age, years_pro):
    if age and years_pro is not None:
        enter_pro = age - years_pro
        return f"{enter_pro}"
    return "N/A"

# Callback to automatically calculate NFL age entered pro league
@app.callback(
    Output('nfl-enter-pro-display', 'children'),
    [Input('age-slider', 'value'),
     Input('nfl-years-pro-slider', 'value')]
)
def calculate_nfl_enter_pro(age, years_pro):
    if age and years_pro is not None:
        enter_pro = age - years_pro
        return f"{enter_pro}"
    return "N/A"
```

4. Model Output and Interactivity

For each sport tab, the dashboard returns a predicted annual salary based on the user's profile and the selected position. Alongside this prediction, the system also displays the calculated BMI and the user's projected age of professional entry. Users can interactively adjust their profile and see how changes in attributes like age or weight impact potential earnings across each sport. This level of interactivity allows for immediate feedback and informed exploration of cross-sport opportunities.



5. Challenges and Resolutions

During the development of this dashboard, several significant challenges emerged, each requiring creative and technical solutions. One of the first issues encountered was the inconsistency in position labels across sports. Each sport has a unique system of classifying positions—for instance, the NFL includes quarterback, wide receiver, and linebacker, whereas the NBA features point guards, shooting guards, and centers. To accommodate these differences, we created sport-specific dropdowns, each with an internally consistent mapping to numerical position labels, ensuring compatibility with the corresponding machine learning models.

A second challenge was the discrepancy in salary reporting formats. While NBA and NFL salaries are listed annually, the FIFA dataset reports wages on a weekly basis. To normalize these values and enable direct comparison across leagues, we converted weekly wages to annual salaries by multiplying by fifty-two. This harmonization improved both model performance and interpretability.

A third obstacle was the absence of professional experience data in the FIFA dataset. Because the Enter_Pro feature relies on this variable, we had to modify our approach specifically for the soccer model. We retrained the FIFA model without this feature, ensuring the model remained structurally sound and predictions remained consistent.

Finally, large model file sizes presented a challenge in terms of application speed and responsiveness. We addressed this by using the joblib library for efficient serialization and loading the models only once when the application starts, instead of reloading them on every interaction. This significantly reduced latency and improved user experience.

6. Insights Gained

The deployment and use of this dashboard offered several valuable insights. In the context of NBA salary prediction, it became evident that players with very similar physical attributes—such as height, weight, and age—can have vastly different salaries. This suggests that intangible factors like basketball IQ, court vision, and individual skill play a more critical role in basketball than pure physique.

In contrast, NFL predictions showed a stronger correlation between physical metrics and salary, especially for positions that demand extreme physicality, such as linemen and running backs. Here, BMI and the age of entering the professional league emerged as significant indicators of earning potential.

For FIFA, although the model used a simplified approach to position encoding, predictions remained surprisingly reliable. This highlights that in soccer, physical and age-related variables can still explain a meaningful portion of salary variance, even without precise role differentiation.

These sport-specific patterns suggest that success and compensation in professional sports depend not just on physical measurements, but also on how each league values specific traits. This insight may prove useful for athletes seeking to evaluate their fit across multiple disciplines.

7. User Experience and Design Principles

The dashboard was designed with three main principles in mind. First, responsiveness was critical. As users change inputs, they receive immediate feedback through updated predictions and recalculated metrics. This real-time interactivity makes the application engaging and educational.

Second, the design prioritizes comparability. Since inputs are shared across sport tabs, users can easily assess how their profile performs in different sports without needing to re-enter data. This structure supports cross-discipline evaluation and encourages experimentation.

Third, the user interface was intentionally kept minimalist. Input fields are logically grouped, labels are clearly defined, and results are displayed in an uncluttered format. These design decisions reduce friction and make the dashboard accessible to a wide audience, regardless of technical background.

8. Future Improvements

To further increase the dashboard's value and applicability, several future enhancements are proposed. One improvement would be to add percentile-based insights, helping users understand how their predicted salary compares to others in the same role or sport. Additionally, incorporating visual elements like histograms or violin plots could provide richer contextual understanding.

Another opportunity lies in the integration of skill-based data. Adding attributes such as sprint speed, agility scores, or shooting accuracy could allow models to better reflect the multifaceted nature of athletic success. Furthermore, enabling the export of prediction results into a downloadable PDF report would allow athletes to share their profiles with coaches or agents.

Lastly, including a localization feature—such as toggling between metric and imperial units—would broaden accessibility for users from different regions. Combined, these improvements would make the dashboard not only more robust but also more personalized and widely usable.

9. Conclusion

This milestone integrates previous data science work into a practical, interactive tool for athlete career planning. By combining predictive modeling, thoughtful UI design, and real-world datasets, the dashboard provides personalized insights that can guide important career decisions.

It stands not just as a technical achievement, but as an example of how machine learning can inform and impact human choices—especially in fields as dynamic and competitive as professional sports.