🚀 Project Report: Predicting Olympic Medals with Machine Learning 🏅

Welcome to the exciting world of data analysis and machine learning! In this project, we embark on a journey to predict the number of medals a country's Olympic team might win based on various factors. Buckle up and get ready for some data-driven fun! 🤓

📚 Project Overview

This Python project aims to analyze historical Olympic data and build a predictive model to forecast the number of medals a team might win. By leveraging powerful libraries like Pandas, Seaborn, and Scikit-learn, we'll explore the relationships between different variables and train a linear regression model to make predictions.

The primary goals of this project are:

1. **Data Exploration**: Gain insights into the Olympic dataset by visualizing relationships between features and target variables.
2. **Feature Engineering**: Preprocess the data by handling missing values, converting data types, and selecting relevant features.
3. **Model Training**: Train a linear regression model using historical data to predict the number of medals a team might win.
4. **Model Evaluation**: Evaluate the performance of the trained model using appropriate metrics and identify areas for improvement.
5. **Prediction**: Use the trained model to make predictions on new, unseen data.

By the end of this project, we'll have a better understanding of how various factors influence a team's medal count and gain hands-on experience in building predictive models using Python.

🔍 Data Exploration

Let's start by taking a peek at the data we're working with:

```
teams = pd.read_csv("teams.csv")
teams =
teams[["team","country","year","athletes","age","prev_medals","medals"]]
teams.fillna(0, inplace=True)
numeric_columns = ['year', 'athletes', 'age', 'prev_medals', 'medals']
teams[numeric_columns] = teams[numeric_columns].astype(int)
```

We've loaded the `teams.csv` file into a Pandas DataFrame and selected the relevant columns for our analysis. We've also handled any missing values by replacing them with zeros, and ensured that our numeric columns have the correct data types.

To better understand the relationships between our variables, we can visualize the data using Seaborn:

```
sns.lmplot(x="athletes", y="medals", data=teams, fit_reg=True, ci=None)
teams.plot.hist(y="medals")
```

These plots give us a visual representation of the relationship between the number of athletes and the number of medals won, as well as the distribution of medals across teams. From these visualizations, we can observe trends and potential correlations that might inform our modeling approach.

🧠 Model Training and Predictions

Now, let's dive into the fun part - training our model and making predictions! 🤖

```
train = teams[teams["year"] < 2012].copy()
test = teams[teams["year"] >= 2012].copy()

reg = LinearRegression()
predictors = ["athletes", "prev_medals"]
target = "medals"

reg.fit(train[predictors], train["medals"])
predictions = reg.predict(test[predictors])
predictions[predictions < 0] = 0
predictions = np.round(predictions)
test["predictions"] = predictions
```

We've split our data into training and testing sets based on the year. We've then trained a linear regression model using the number of athletes and previous medals as predictors, and the number of medals as the target variable.

The linear regression algorithm assumes a linear relationship between the input features (predictors) and the target variable. By training the model on historical data, it learns the coefficients that best fit the linear equation, allowing us to make predictions on new, unseen data.

To ensure our predictions are meaningful, we've capped any negative predictions to zero (since medal counts can't be negative) and rounded them to the nearest integer.

📊 Model Evaluation

To evaluate our model's performance, we can calculate the mean absolute error:

```
error = mean_absolute_error(test["medals"], test["predictions"])
```

```
errors = (test["medals"] - test["predictions"]).abs()
```

The mean absolute error (MAE) gives us an idea of how far off our predictions are from the actual medal counts, on average. A lower MAE indicates better predictive performance.

We can also visualize the errors to identify any patterns or outliers:

```
errors.plot.hist()
```

```
errors.plot.box()
```

These visualizations can help us understand the distribution of errors and identify any potential biases or skewness in our predictions.

🔍 Feature Importance and Limitations

While the current model uses the number of athletes and previous medals as predictors, we could potentially improve its performance by incorporating additional relevant features. Some examples of features that might influence medal counts include:

- Funding and resources allocated to the Olympic team
- Population size and demographics of the country
- Historical performance and consistency in specific sports
- Coaching and training facilities

However, it's important to note that this model, like any predictive model, has limitations. Factors such as political situations, unforeseen events, or changes in competition rules can significantly impact medal counts in ways that may not be captured by historical data alone.

🏆 Conclusion

Through this project, we've explored the fascinating world of data analysis and machine learning in the context of Olympic medal predictions. We've gained insights into the relationships between various factors and the number of medals won, and we've built a predictive model to forecast future medal counts.

While our model may not be perfect, it serves as a valuable learning experience and a stepping stone towards more advanced techniques in the field of data science. By continuously iterating, incorporating new features, and exploring alternative modeling approaches, we can improve our predictive capabilities.

Remember, the journey is just as important as the destination. So keep learning, keep exploring, and most importantly, have fun with data! 🎉

If you have any further questions or would like to discuss potential improvements or extensions to this project, feel free to reach out. I'll be more than happy to provide additional insights and guidance. 🙌