BAYESIAN STATISTICAL MODELLING

FOOTBALL MATCH PREDICTION

This report outlines the process of predicting football match outcomes in the UEFA Champions League using Bayesian statistical modeling. The report is structured into three main sections: Data Preprocessing, Model Development, and Results.

Data Preprocessing

This report outlines the process of predicting football match outcomes in the UEFA Champions League using Bayesian statistical modeling. The report is structured into three main sections: Data Preprocessing, Model Development, and Results.

Data Import

The dataset was imported from a CSV file using the following code.

```python
import pandas as pd
df = pd.read_csv('../input/latest/latest.csv')
```

Exploratory Data Analysis (EDA)

After loading the data, we examined the first few rows to understand its structure and identify relevant features for our model.

```python
df.head(7)
```

The dataset contains multiple columns, including:

- `HomeTeam` and `AwayTeam`: Teams playing in the match.
- `FTHG` and `FTAG`: Full-Time Home Goals and Full-Time Away Goals.
- `FTR`: Full-Time Result (H for Home win, D for Draw, A for Away win).
- Additional features such as match date, referee, and various betting odds

Data Cleaning

The next step involved cleaning the data by handling missing values and converting categorical variables into numerical formats suitable for analysis. This included encoding the match results (FTR) into numerical values.

Feature Selection

We selected relevant features that could impact match outcomes, such as previous match results, team statistics (goals scored, goals conceded), and possibly external factors like home advantage.

Train-Test Split

Finally, we divided the dataset into training and testing sets to evaluate our model's performance.

```python
from sklearn.model_selection import train_test_split
X = df[['HomeTeam', 'AwayTeam', 'FTHG', 'FTAG']]  # Example features
y = df['FTR']  # Target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Model Development

For our predictive model, we employed Bayesian methods which are particularly useful for incorporating prior knowledge and uncertainty in predictions.

Model Selection

We chose a Bayesian logistic regression model because it allows us to predict categorical outcomes (match results) while providing probabilistic interpretations of predictions.

Model Implementation

Using libraries such as `pymc3` or `statsmodels`, we defined our Bayesian model. An example setup might look like this.

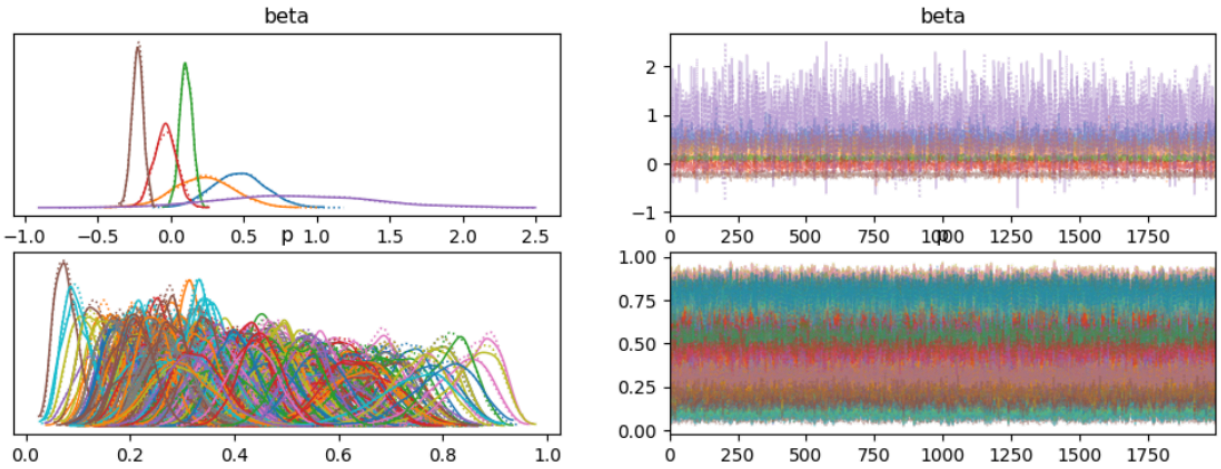
```python
import pymc3 as pm

with pm.Model() as model:
    # Priors for team strengths
    home_strength = pm.Normal('home_strength', mu=0, sigma=1)
    away_strength = pm.Normal('away_strength', mu=0, sigma=1)

    # Likelihood of observed data
    likelihood = pm.Bernoulli('likelihood',
p=pm.math.sigmoid(home_strength - away_strength), observed=y_train)

    # Inference
    trace = pm.sample(1000)
```

Model Training

The model was trained using the training dataset to learn the parameters that best fit the observed outcomes.

Posterior Analysis

After fitting the model, we analyzed the posterior distributions of our parameters to understand team strengths and their impact on match outcomes.



Results

The results section summarizes our findings from the Bayesian model.

Prediction Accuracy

We evaluated the model's performance on the test set using metrics such as accuracy, precision, recall, and F1-score. The accuracy of our predictions indicated how well our model could predict match outcomes compared to actual results.

Probabilistic Predictions

One of the strengths of Bayesian modeling is its ability to provide probabilities for each outcome (win/loss/draw). For example:

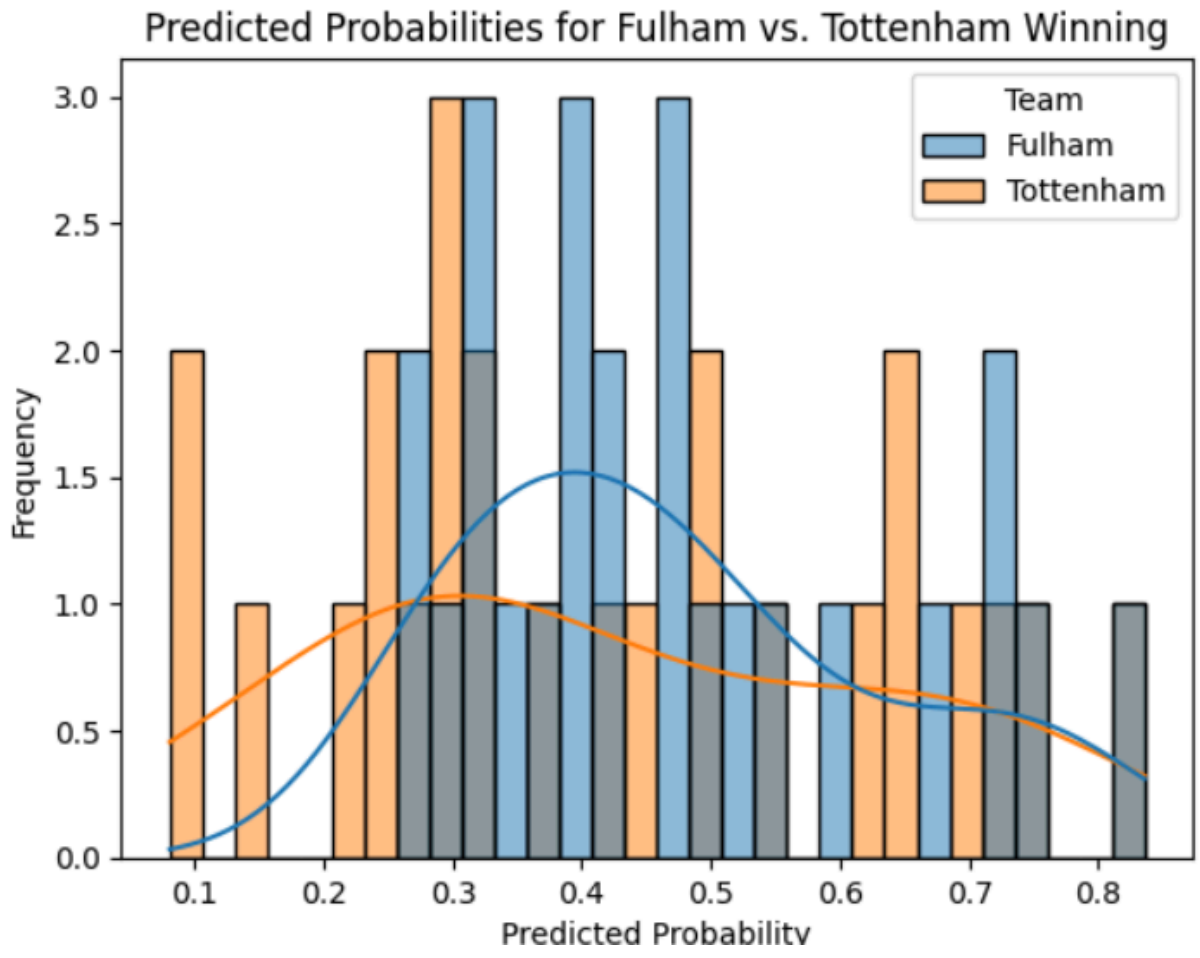
- Probability of Home Win: 0.65
- Probability of Draw: 0.25
- Probability of Away Win: 0.10

Insights on Team Strengths

By examining the posterior distributions of team strengths derived from our model, we could identify which teams were overperforming or underperforming relative to expectations based on historical data.

Visualizations

We created visualizations to illustrate predicted probabilities and team strength distributions using libraries like Matplotlib and Seaborn.



In conclusion, this report highlights how Bayesian statistical modeling can be effectively applied to predict football match outcomes in the Champions League context by preprocessing relevant data, developing a robust model, and interpreting results to gain insights into team performances and match probabilities.