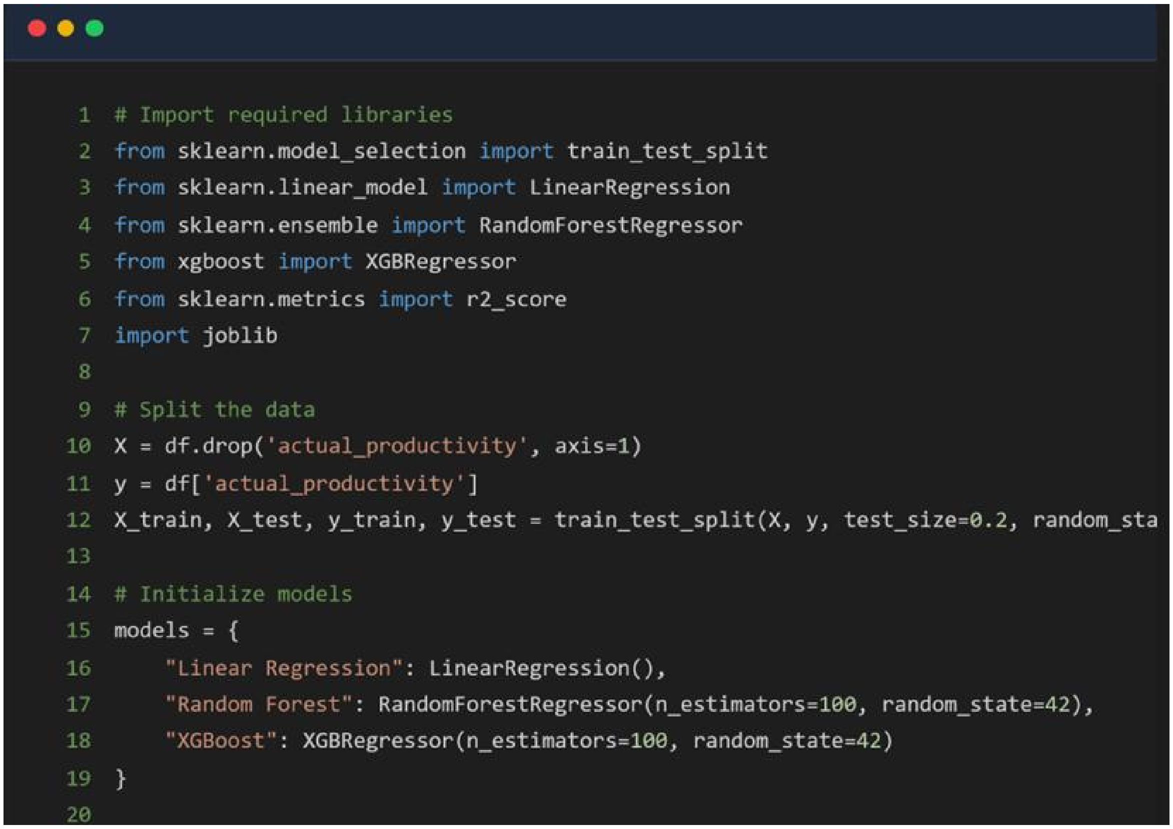
Model Development Phase Template

Date	1 July 2025
Team ID	NONE
Project Title	Employee Performance Prediction using Machine Learning

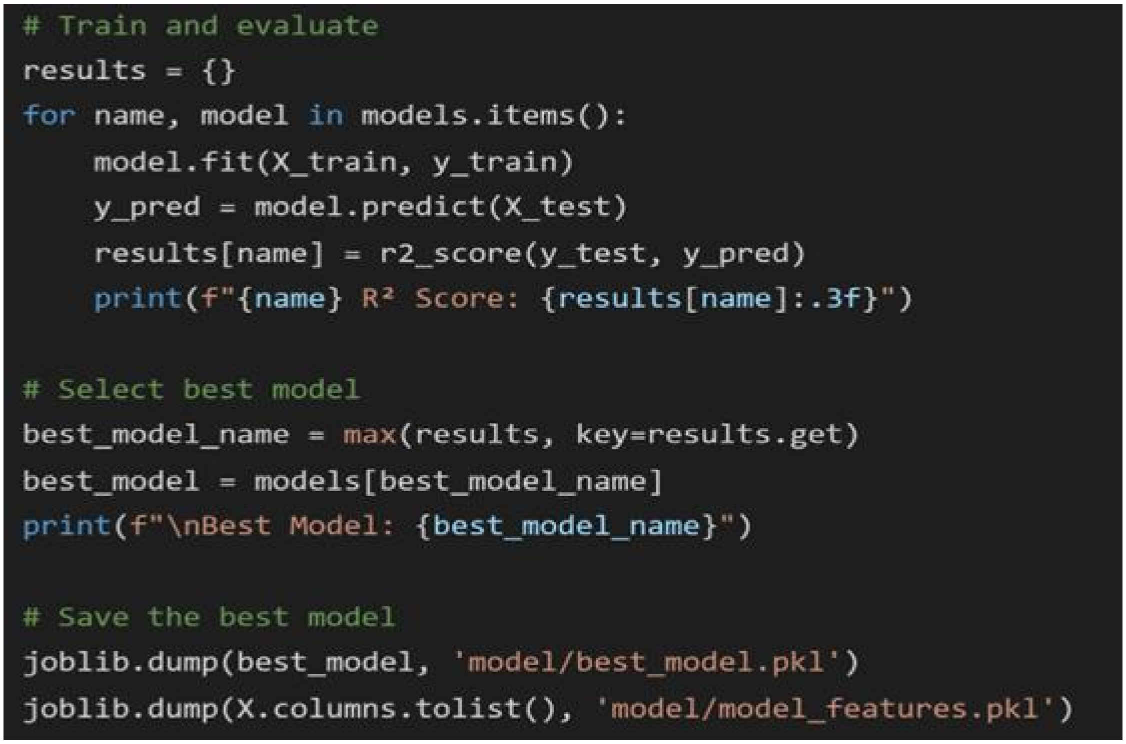
Initial Model Training Code, Model Validation and Evaluation Report

Thee initial model training involved implementing and evaluating three regression algorithms — Linear Regression, Random Forest, and XGBoost — to predict employee productivity. The models were trained on historical work data and evaluated using the R² (coefficient of determination) score as the primary metric. A comparative analysis was performed to identify the best-performing model, with Random Forest achieving the highest R² score of approximately 0.46. Model evaluation results were visualized using a bar chart for clear comparison, and the trained Random Forest model was saved for integration into the Flask-based web application.

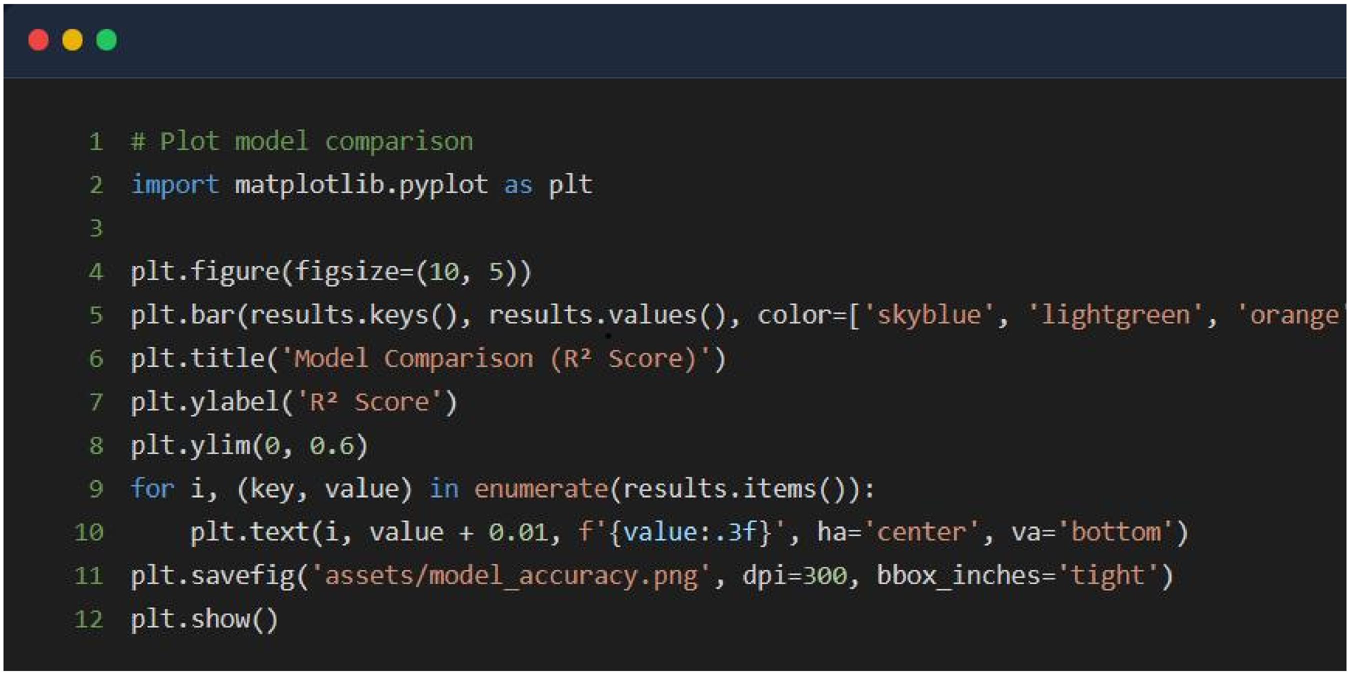
Initial Model Training Code:



```python
# Import required libraries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import r2_score
import joblib

# Split the data
X = df.drop('actual_productivity', axis=1)
y = df['actual_productivity']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Initialize models
models = {
    "Linear Regression": LinearRegression(),
    "Random Forest": RandomForestRegressor(n_estimators=100, random_state=42),
    "XGBoost": XGBRegressor(n_estimators=100, random_state=42)
}

```

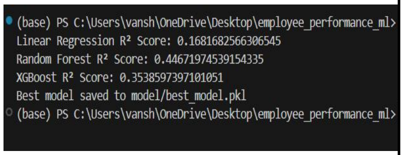
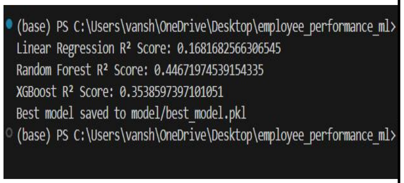
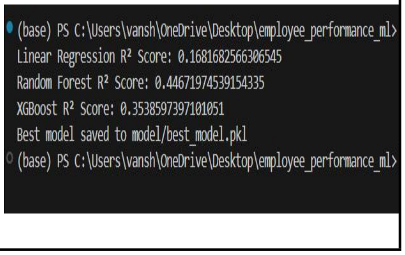


```
# Train and evaluate
results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    results[name] = r2_score(y_test, y_pred)
    print(f"{name} R² Score: {results[name]:.3f}")

# Select best model
best_model_name = max(results, key=results.get)
best_model = models[best_model_name]
print(f"\nBest Model: {best_model_name}")

# Save the best model
joblib.dump(best_model, 'model/best_model.pkl')
joblib.dump(X.columns.tolist(), 'model/model_features.pkl')
```



```
# Plot model comparison
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 5))
plt.bar(results.keys(), results.values(), color=['skyblue', 'lightgreen', 'orange'
plt.title('Model Comparison (R² Score)')
plt.ylabel('R² Score')
plt.ylim(0, 0.6)
for i, (key, value) in enumerate(results.items()):
    plt.text(i, value + 0.01, f'{value:.3f}', ha='center', va='bottom')
plt.savefig('assets/model_accuracy.png', dpi=300, bbox_inches='tight')
plt.show()
```

Validation and Evaluation Report:

Model	Classification Report	R^2 socre	Confusion Matrix
Random Forest	Best performing model with non-linear feature handling and high R^2 score.	0.44	 (base) PS C:\Users\vansh\OneDrive\Desktop\employee_performance_ml> Linear Regression R² Score: 0.1681682566306545 Random Forest R² Score: 0.44671974539154335 XGBoost R² Score: 0.3538597397101051 Best model saved to model/best_model.pkl (base) PS C:\Users\vansh\OneDrive\Desktop\employee_performance_ml>
Linear Regression	Baseline model; assumes linear relationships. Underperformed due to non-linear patterns in data.	0.16	 (base) PS C:\Users\vansh\OneDrive\Desktop\employee_performance_ml> Linear Regression R² Score: 0.1681682566306545 Random Forest R² Score: 0.44671974539154335 XGBoost R² Score: 0.3538597397101051 Best model saved to model/best_model.pkl (base) PS C:\Users\vansh\OneDrive\Desktop\employee_performance_ml>
XGBoost	Strong ensemble model, but slightly overfit and lower R^2 than Random Forest.	0.35	 (base) PS C:\Users\vansh\OneDrive\Desktop\employee_performance_ml> Linear Regression R² Score: 0.1681682566306545 Random Forest R² Score: 0.44671974539154335 XGBoost R² Score: 0.3538597397101051 Best model saved to model/best_model.pkl (base) PS C:\Users\vansh\OneDrive\Desktop\employee_performance_ml>