```
#module7
```

Module 7: Model Development and Evaluation

Name: Srinivasa Reddy Julakanti Registration Number: 21BDS0220

```
# Import necessary libraries
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
import numpy as np
import pandas as pd

# Load the dataset
data = pd.read_csv('bank_marketing_test.csv')
data.head(5)
```

housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome
no	yes	telephone	may	mon	...	1	999	0	nonexistent
no	no	telephone	may	mon	...	1	999	0	nonexistent
yes	no	telephone	may	mon	...	1	999	0	nonexistent
yes	no	telephone	may	mon	...	1	999	0	nonexistent
yes	yes	telephone	may	mon	...	1	999	0	nonexistent

```
print(data.columns)
```

```
Index(['age', 'duration', 'campaign', 'pdays', 'previous', 'emp.var.rate',
       'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed',
       'job_blue-collar', 'job_entrepreneur', 'job_housemaid',
       'job_management', 'job_retired', 'job_self-employed', 'job_services',
       'job_student', 'job_technician', 'job_unemployed', 'job_unknown',
       'marital_married', 'marital_single', 'marital_unknown',
       'education_basic.6y', 'education_basic.9y', 'education_high.school',
       'education_illiterate', 'education_professional.course',
       'education_university.degree', 'education_unknown', 'default_unknown',
       'housing_unknown', 'housing_yes', 'loan_unknown', 'loan_yes',
       'contact_telephone', 'month_aug', 'month_dec', 'month_jul', 'month_jun',
       'month_mar', 'month_may', 'month_nov', 'month_oct', 'month_sep',
       'day_of_week_mon', 'day_of_week_thu', 'day_of_week_tue',
       'day_of_week_wed', 'poutcome_nonexistent', 'poutcome_success', 'y_yes'],
      dtype='object')
```

```
X = data.drop(columns=['y_yes'])  # All features except target
y = data['y_yes']  # Target variable (converted to numerical)
```

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Initialize and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate performance metrics
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mse)
accuracy = np.mean(np.abs(y_pred - y_test) <= 0.5)  # Accuracy within 0.5 tolerance

# Display results
print(f"Mean Squared Error (MSE): {mse}")
print(f"R-squared (R²): {r2}")
print(f"Mean Absolute Error (MAE): {mae}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"Accuracy (within 0.5 tolerance): {accuracy}")
```

```
Mean Squared Error (MSE): 0.06680864832691553
R-squared (R²): 0.3133292642080562
Mean Absolute Error (MAE): 0.15289686643350686
Root Mean Squared Error (RMSE): 0.25847368981564744
Accuracy (within 0.5 tolerance): 0.9089805825242718
```