Rasmus Dyrby Damgaard Machine Learning ITA22
11-06-2025

Exam Score Predictor Chatbot

1. Introduction

For this exam project, I developed a chatbot that uses machine learning to predict the exam score of students, based on a number of different inputs. These inputs vary from “study hours” to "Netflix hours”. The user enters information in a command-line (CMD) style bot, such as how many hours they study a day, sleep, what diet etc. The chatbot will then provide an estimated exam score, as well as feedback of sentiment score, and a brief Wordcloud summary of the conversation.

2. Data and Pre-processing

The data is based on a synthetic dataset with approximately 1000 students, stored as a CSV file. Features include:

age, gender, study hours per day, social media hours, Netflix hours, part-time job, attendance percentage, sleep hours, diet quality, exercise frequency, parental education level, internet quality, mental health rating, extracurricular participation, and the actual exam score.

The categorical variables were encoded using OneHotEncoder, that transformed the categories into separate binary columns. This allows the machine learning algorithm to interpret them numerically.

For the model evaluation, the data was split into the classic 80/20 percentage. 80% for training, and 20% for testing.

3. Methods and Machine Learning Algorithms

I built predictive models using RandomForestRegressor and LinearRegression, to compare performance between the two.

```
model_choice = input(
    "Which exam score prediction model do you want to use?\n" +
    "Type 'rf' for RandomForest, 'lr' for LinearRegression: "
).strip().lower()
if model_choice == "lr":
    print("You chose Linear Regression.")
    model = joblib.load("student_score_linearregression.pkl")
else:
    print("Using Random Forest (default).")
    model = joblib.load("student_score_randomforest.pkl")
```

During training and evaluation, I used RMSE (Root Mean Squared Error) as the primary performance metric.

The chatbot uses lemmatization via spaCy, to reduce words to their base form, making it easier to match the input to the expected features. It then extracts the numerical value and category for each relevant entity, for example “study hours”, to build the input for the model.

4. Experiments and Results

4.1 Model Performance

To analyze the performance of my two algorithms, I measured using Root Mean Squared Error (RMSE), which measures the average prediction errors. In general, the lower RMSE, the better accuracy of a model.

Below is the RMSE for both models:

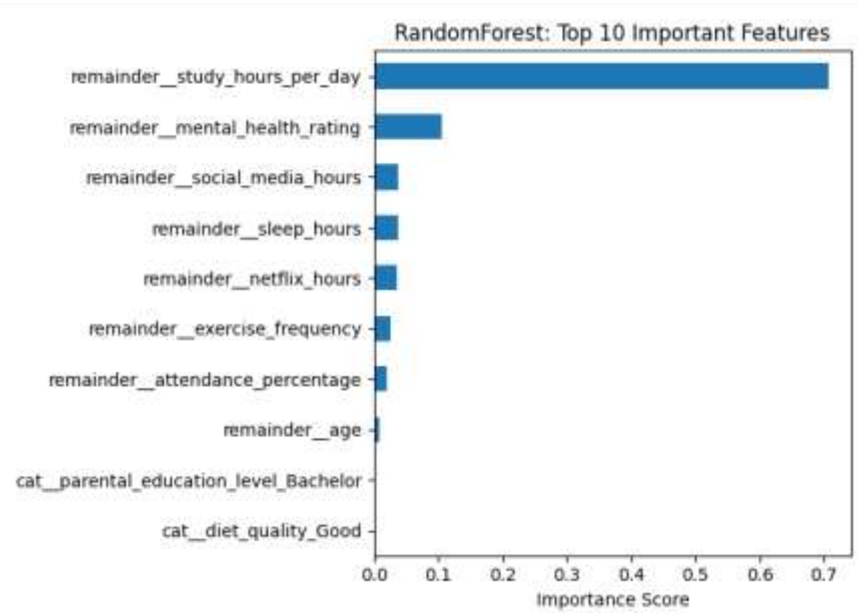
```
Model comparison (Train vs. Test RMSE):
RandomForest        : Train RMSE = 2.42, Test RMSE = 6.29
LinearRegression    : Train RMSE = 5.34, Test RMSE = 5.15
```

In my project, RandomForest got a very low training RMSE, 2.42, but a higher Test RMSE,6.29. This indicates signs of overfitting, due to it performing very well on training data, but much worse on unseen data. However, LinearRegression has very similar training and test RMSE values, respectively 5.34 and 5.15, suggesting no signs of overfitting. The performance on the training set is not as strong as RandomForest, but it generalizes better to new data. These values are relatively low, when considering that the exam score ranges from 0 to 100.

LinearRegression achieved the lowest RMSE score for the test data and will therefore be the best choice for chatting.

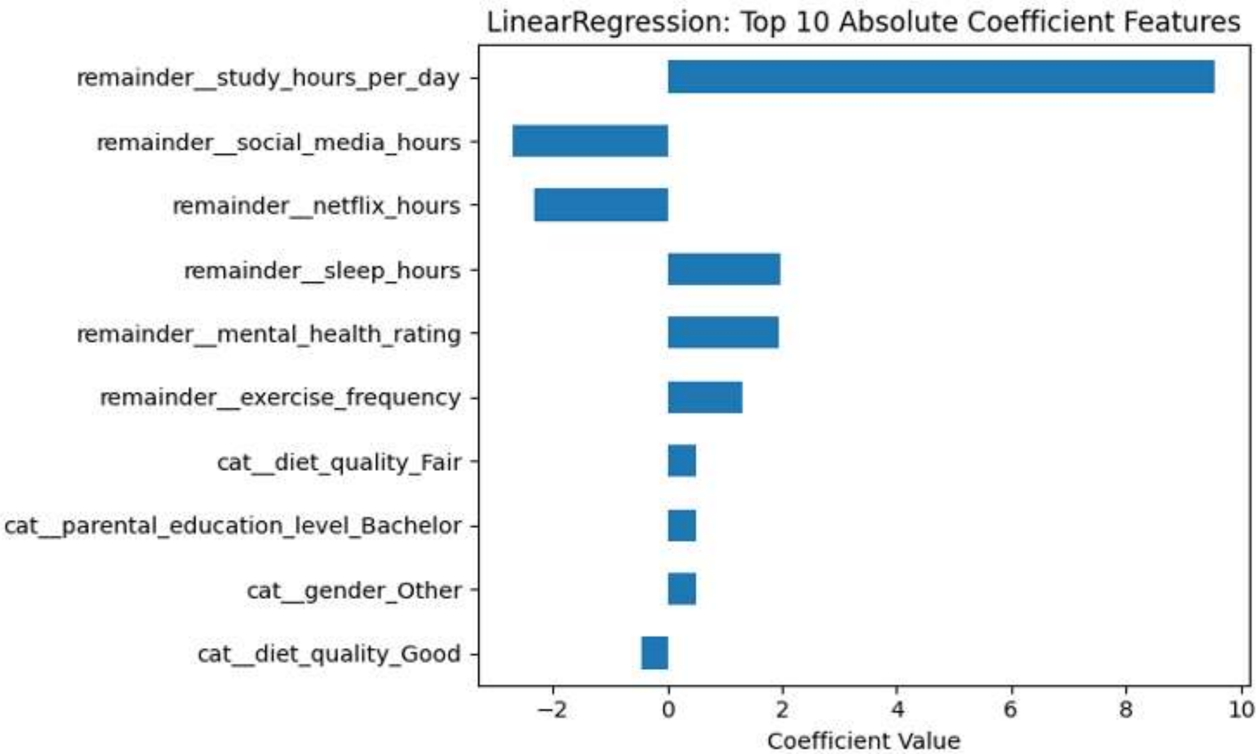
4.2 Feature Importance

RandomForest:



Here we see the top 10 most important features, that influence the exam score predictions. As one could imagine, the "study hours per day" feature is the most important by a long way.

LinearRegression coefficients:



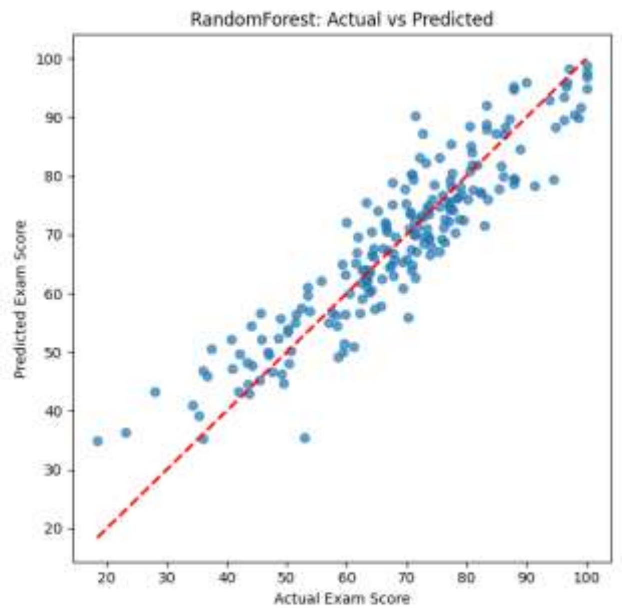
Here we see the top 10 most influential features in the LinearRegression model. The most impactful feature is still "study hours per day". We can also see that "social media hours", has a negative effect on the exam score, which again makes quite good sense.

Difference between Feature Importance in RandomForest and LinearRegression:

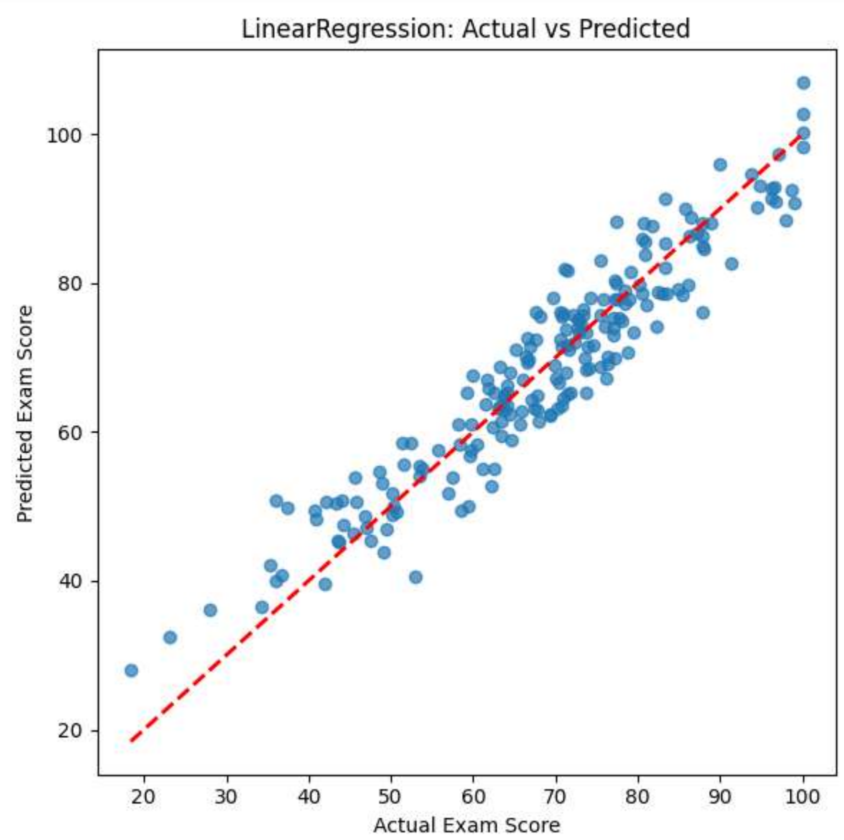
The main difference is that in LinearRegression, you can see exactly how much each factor pulls the exam score up or down, and by how much. For example, a negative value on "social media hours" means a lower score. Whereas in RandomForest, the important features are those that help the model make better predictions – it does not tell you if the effect is positive or negative.

4.3 Actual vs. Predicted

RandomForest:



LinearRegression:



As we can see, in both of these “actual vs predicted” scatter plots, most points cluster around the diagonal red line (the red line represents perfect predictions). LinearRegression is slightly closer to the red line, which reflects the lower RMSE. Overall, both models performed well, but LinearRegression has a better performance.

4.4 Scenario Testing (Chatbot Output)

SCENARIO	INPUT	EXPECTED SCORE	**MODEL OUTPUT**	SENTIMENT SCORE	EXAMPLE (SEE ATTACHMENTS)
DOES VERY LITTLE	See Example1	Low	**24.50**	-0.1	Example1
DOES A LOT	See Example2	High	**97.18**	0.11	Example2
DOES MEDIUM	See example3	Medium	**60.73**	0.11	Example3

This demonstrates that the chatbot and model react sensibly to different inputs, and different student habits.

The 3 examples can be seen under Attachments.

6. Implementation details:

Imputation: To handle missing inputs, I automatically imputed missing values using mean/mode from the training data.

Creating the mean/mode value and dumping the values:

```
impute_vals = {}
for col in X.columns:
    if X[col].dtype in ("float64", "int64"):
        impute_vals[col] = X[col].mean()
    else:
        impute_vals[col] = X[col].mode()[0]
dump(impute_vals, "impute_defaults.pkl")
```

Importing the values:

```
impute_vals = joblib.load("impute_defaults.pkl")
```

When making the prediction, it fills the missing inputs with the corresponding mean/mode value:

```
def get_prediction():
    input_vals = {}
    for f in features:
        val = features[f]
        if val is None:
            val = impute_vals[f]
        input_vals[f] = val
    input_df = pd.DataFrame([input_vals])
    pred = model.predict(input_df)[0]
    return max(0, min(pred, 100))
```

Clipping: As regression models can produce unrealistic scores (e.g., above 100), I implemented clipping to ensure predictions stay between 0 and 100.

Last line of "Get_predicitons()":

```
return max(0, min(pred, 100))
```

This ensures that the chatbot only shows "exam scores" between 0 and 100.

User warnings: The chatbot notifies the user if required fields are missing, making it clear that prediction accuracy might be affected.

Sentiment & conversation summary: The chatbot automatically creates a word cloud and sentiment estimate based on the conversation.

Each message from the user will append the sentiment polarity, using TextBlob, to a list:

```
def parse_input(user_input):
    all_text.append(user_input)
    sentiment_scores.append(TextBlob(user_input).sentiment.polarity)
```

In Summarize_conversation(), it will calculate the average sentiment score and print the value with a corresponding message.

```
def summarize_conversation():
    if sentiment_scores:
        avg_sentiment = sum(sentiment_scores) / len(sentiment_scores)
        print(f"Overall sentiment score: {avg_sentiment:.2f}")
        if avg_sentiment > 0.1:
            print("Your overall mood seems to be quite positive")
        elif avg_sentiment < -0.1:
            print("It seems like you're having a tough time. Take care")
        else:
            print("Your mood appears to be neutral.")
```

Input parsing limitation: Because of the way the chatbot extracts feature values, the user must provide exactly one feature per input line/message. If multiple features and values are provided in one message, the chatbot may incorrectly assign values, because the chatbot always matches the first relevant number for each type of feature.

Model selection: As seen previously, the user can select which ML model (RandomForest or LinearRegression) to use from the start of the chatbot.

Future improvements: for better predicted scores, the user could be required to give a minimum number of filled values, before the chatbot can make the prediction. The input parsing limitation could be improved, so there could be more features in one sentence. This would make the chatbot more user-friendly.

7. Conclusion

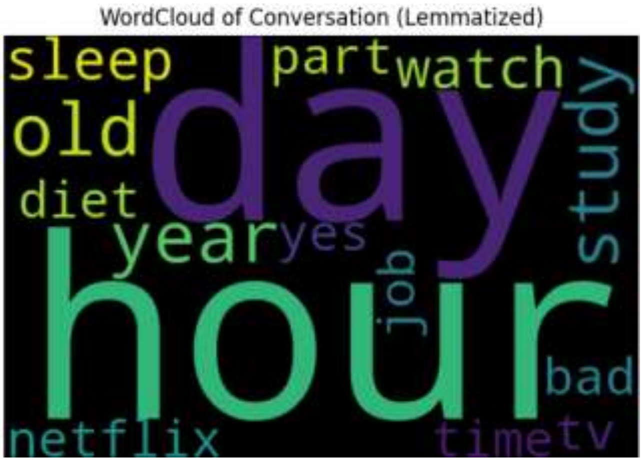
This Chatbot project shows how a simple chatbot can use machine learning to predict exam scores for students, based on user input regarding study habits, lifestyle habits etc. Both Machine Learning models gave reasonable results, where LinearRegression performed slightly better. The chatbot can handle missing data, ensure predictions remain relevant and not going out of the 0-100 score. It gives helpful feedback, sentiment analysis and shows a word cloud of the message.

To see entire project without Virtual Environment, this includes CSV and the PKL files follow this link:

https://github.com/rasmusdyrby/Machine_learning_exam

9. Attachments

9.1: example1:



9.2: example2:

```
Which exam score prediction model do you want to use?
Type 'rf' for RandomForest, 'lr' for LinearRegression: lr
You chose Linear Regression.
Welcome to the Student Exam Score Predictor Bot!
You can start by telling me things like your age, study habits, sleep hours, etc.
***Please type numbers (1,2,3) and follow with 'hours' afterwards***
Type 'exit' to quit and get your score prediction and summary.

You: I am 24 years old
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, study_hours_per_day, social_media_hours, netflix_hours, part_time_job, attendance_percentage, sleep_hours, diet_quality, exercise_frequency, parental_education_level, internet_
quality, mental_health_rating, extracurricular_participation?
You: I study 6 hours a day
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, netflix_hours, part_time_job, attendance_percentage, sleep_hours, diet_quality, exercise_frequency, parental_education_level, internet_quality, mental_healt
h_rating, extracurricular_participation?
You: I sleep 8 hours a day
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, netflix_hours, part_time_job, attendance_percentage, diet_quality, exercise_frequency, parental_education_level, internet_quality, mental_health_rating, ext
racurricular_participation?
You: I watch netflix and tv 1 hour a day
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, part_time_job, attendance_percentage, diet_quality, exercise_frequency, parental_education_level, internet_quality, mental_health_rating, extracurricular_pa
rticipation?
You: I have a good diet
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, part_time_job, attendance_percentage, exercise_frequency, parental_education_level, internet_quality, mental_health_rating, extracurricular_participation?
You: no I dont have a part time job
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, attendance_percentage, exercise_frequency, parental_education_level, internet_quality, mental_health_rating, extracurricular_participation?
You: I exercise 3 times a week
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, attendance_percentage, parental_education_level, internet_quality, mental_health_rating, extracurricular_participation?
You: exit
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, attendance_percentage, parental_education_level, internet_quality, mental_health_rating, extracurricular_participation?
You: exit

Bot: Note, you didn't provide these: gender, social_media_hours, attendance_percentage, parental_education_level, internet_quality, mental_health_rating, extracurricular_participation.
The prediction may be less accurate as a result.

 Predicted Exam Score: 97.18
Overall sentiment score: 0.10
Your mood appears to be neutral.
```

WordCloud of Conversation (Lemmatized)

9.3: example3:

```
Which exam score prediction model do you want to use?
Type 'rf' for RandomForest, 'lr' for LinearRegression: lr
You chose Linear Regression.
Welcome to the Student Exam Score Predictor Bot!
You can start by telling me things like your age, study habits, sleep hours, etc.
***Please type numbers (1,2,3) and follow with 'hours' afterwards***
Type 'exit' to quit and get your score prediction and summary.

You: I am 26 years old
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, study_hours_per_day, social_media_hours, netflix_hours, part_time_job, attendance_percentage, sleep_hours, diet_quality, exercise_frequency, parental_education_level, internet_
quality, mental_health_rating, extracurricular_participation?
You: I study 3 hours a day
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, netflix_hours, part_time_job, attendance_percentage, sleep_hours, diet_quality, exercise_frequency, parental_education_level, internet_quality, mental_healt
h_rating, extracurricular_participation?
You: I sleep 6 hours a day
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, netflix_hours, part_time_job, attendance_percentage, diet_quality, exercise_frequency, parental_education_level, internet_quality, mental_health_rating, ext
racurricular_participation?
You: I watch netflix 2 hours a day
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, part_time_job, attendance_percentage, diet_quality, exercise_frequency, parental_education_level, internet_quality, mental_health_rating, extracurricular_pa
rticipation?
You: I have a fair diet
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, part_time_job, attendance_percentage, exercise_frequency, parental_education_level, internet_quality, mental_health_rating, extracurricular_participation?
You: i exercise 1 time a week
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, part_time_job, attendance_percentage, parental_education_level, internet_quality, mental_health_rating, extracurricular_participation?
You: no i dont have a part time job
Bot: Thanks! Got that.
Bot: Could you also tell me about: gender, social_media_hours, attendance_percentage, parental_education_level, internet_quality, mental_health_rating, extracurricular_participation?
You: exit

Bot: Note, you didn't provide these: gender, social_media_hours, attendance_percentage, parental_education_level, internet_quality, mental_health_rating, extracurricular_participation.
The prediction may be less accurate as a result.

 Predicted Exam Score: 68.73
Overall sentiment score: 0.11
```

WordCloud of Conversation (Lemmatized)