Corporate credit risk classification

JAN ABDULLAH

Overview

- This project takes a supervised machine learning approach to predicting the credit risk of corporate bonds, using credit ratings as labels.
- The model has 3 target risk classes: low, medium, and high
- Why is credit risk important?
  - Cost of capital for corporations
  - Rate of return for investors

Obtaining Data

- We use a corporate credit ratings dataset found on Kaggle, and enrich the dataset by appending reference features from other sources as well.

Scrubbing Data

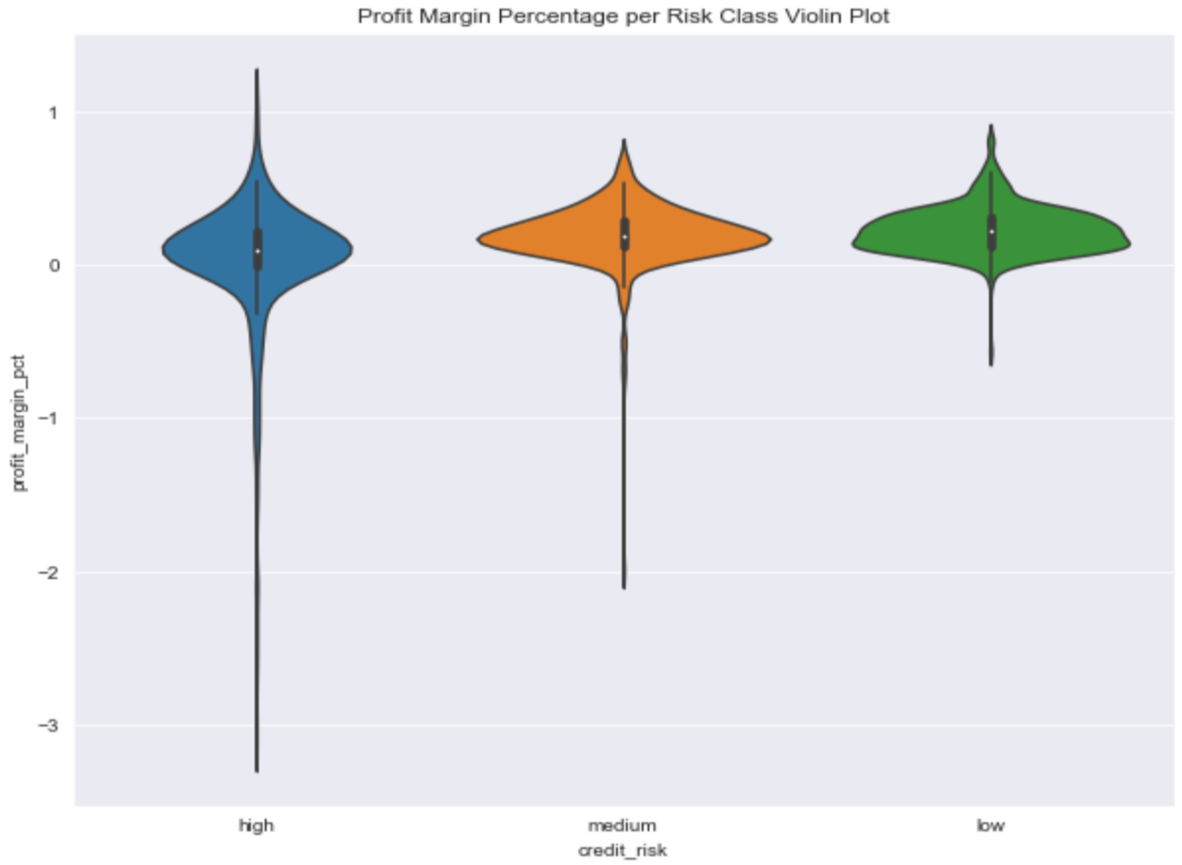
- We clean the data on a risk class basis, by splitting the dataset by each class first and then recombining after cleaning. This should promote clearer distinction between classes.
- We also remove all data points dated before 2010.
  - Why?
    - Questionable ratings agencies practices leading up to 2008 recession.

Explore Data

- We perform some data integrity checks by looking at patterns in some of the features and validating it with some common knowledge on credit risk determination.
- Note that although we only performed EDA on a few features at a time, a lot of other factors are always involved in credit risk determination.

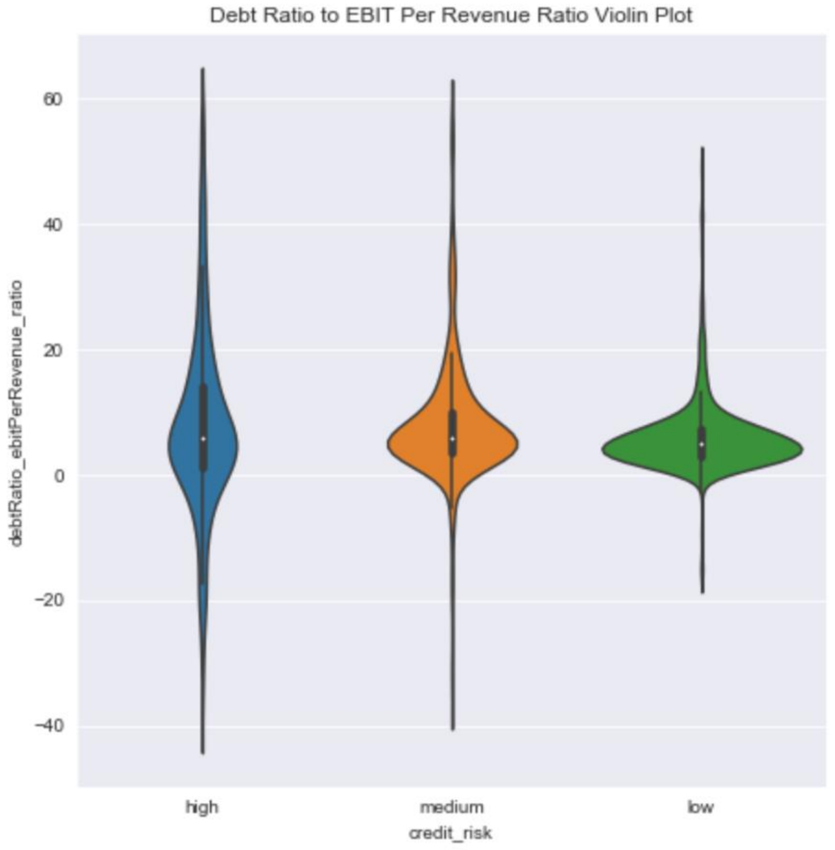
Explore Data (Cont.)

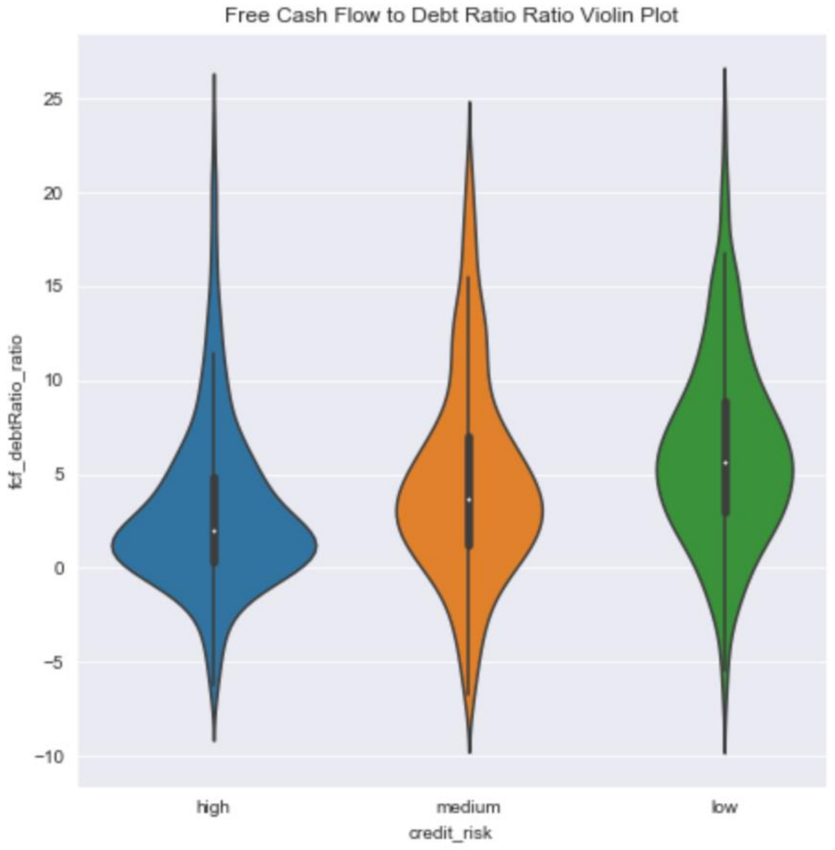
- **Question 1: What is the relationship between net profit percentage and credit risk?**



Explore Data (Cont.)

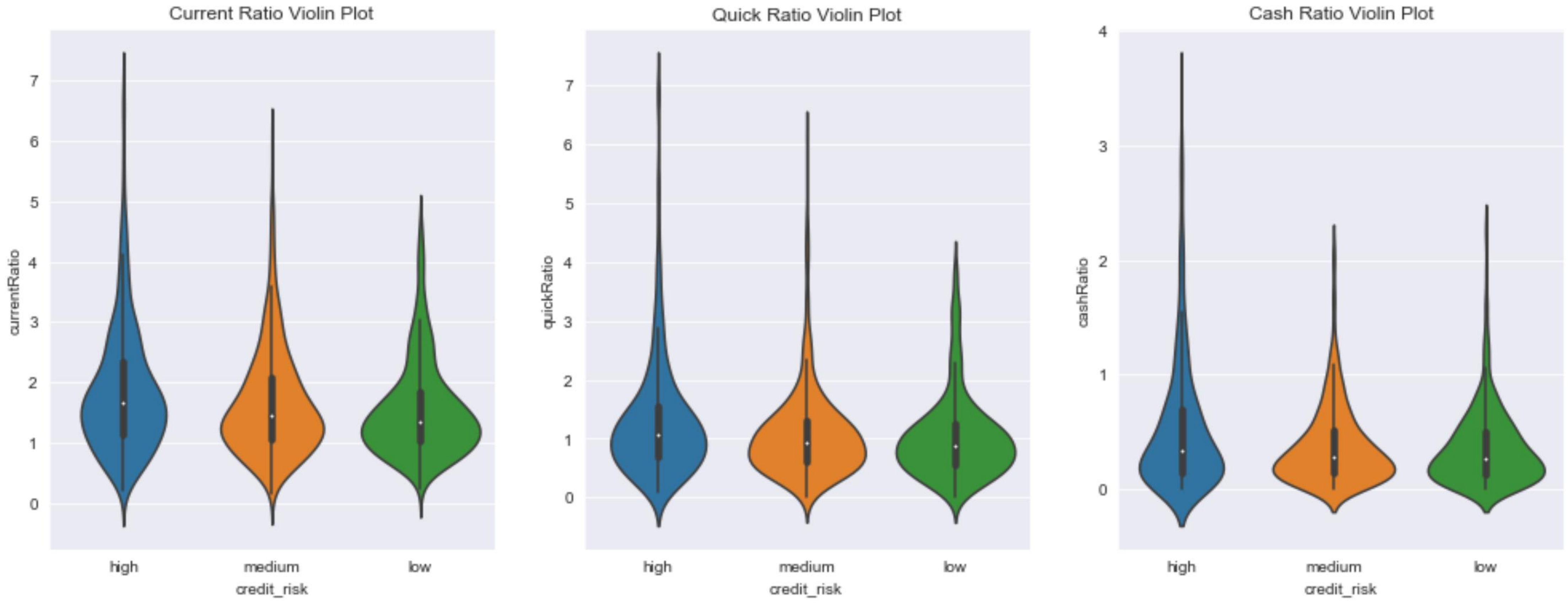
- **Question 2: What are the relationships of key debt ratios to credit risk levels in our dataset?**





Explore Data (Cont.)

- **Question 3: What are the relationships between key liquidity ratios and credit risk levels in our dataset?**

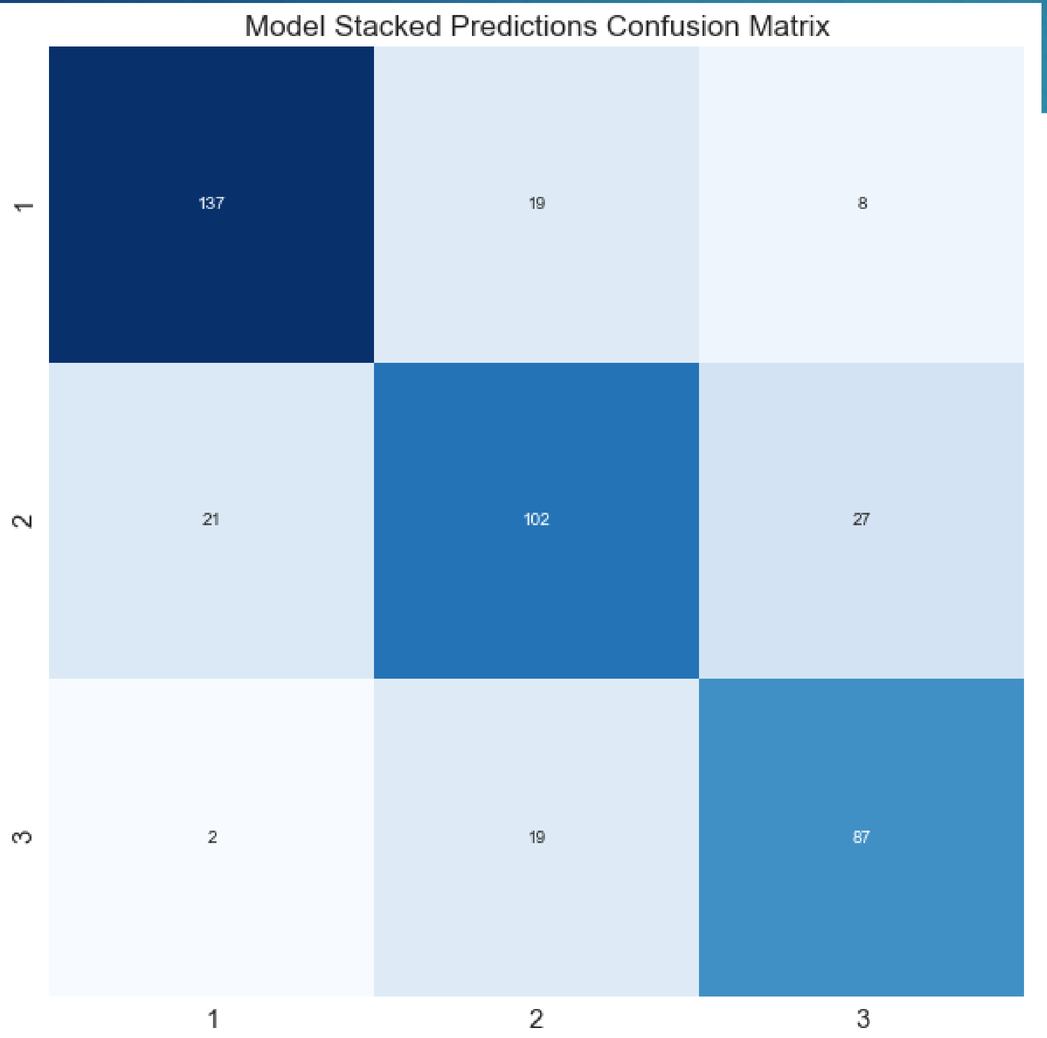


Modeling

- We stack a few different parameter-tuned models to generate our final model. The sklearn classifiers in the stack are:
  1. Bagging
  2. Random Forest
  3. Gradient Boosting
  4. XGBoost

  The stacked model score is 77.25%.

Modeling (Cont.)



Key:

1 – high risk
2 – medium risk
3 – low risk

Interpretability

- Model does fairly well in making class predictions.
- The error of predicting a “low” risk class when the actual is “high”, or vice versa, is 2.37%, which is quite good.
- The rest of the errors are understandable. In cases where risk levels are on the borders between classes, the model can predict these incorrectly.

Future Work

- Invest more time and resources into:
  - Better hyper-parameter tuning
  - Better cleaning configurations
  - Better preprocessing configurations

Thank you!