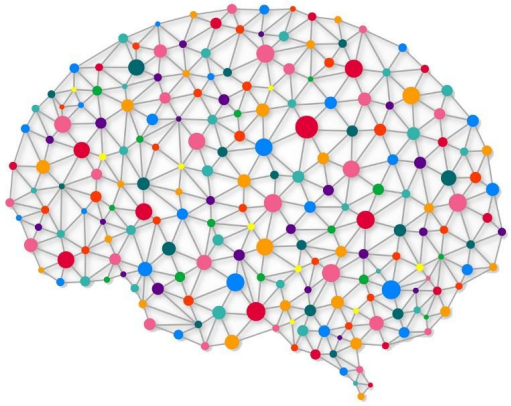
Bug Prediction with Neural Nets

Using regression- and classification-based approaches

Bachelor Thesis
06.02.2018



Sébastien Broggi
Haidar Osman
Prof. Dr. Oscar Nierstrasz

Motivation

I. **Bug prediction on plain text**

Statistical code analysis + different machine learning algorithms already used

Vectorization of text as new approach

II. **Bug prediction with code features**

Different machine learning algorithms already used

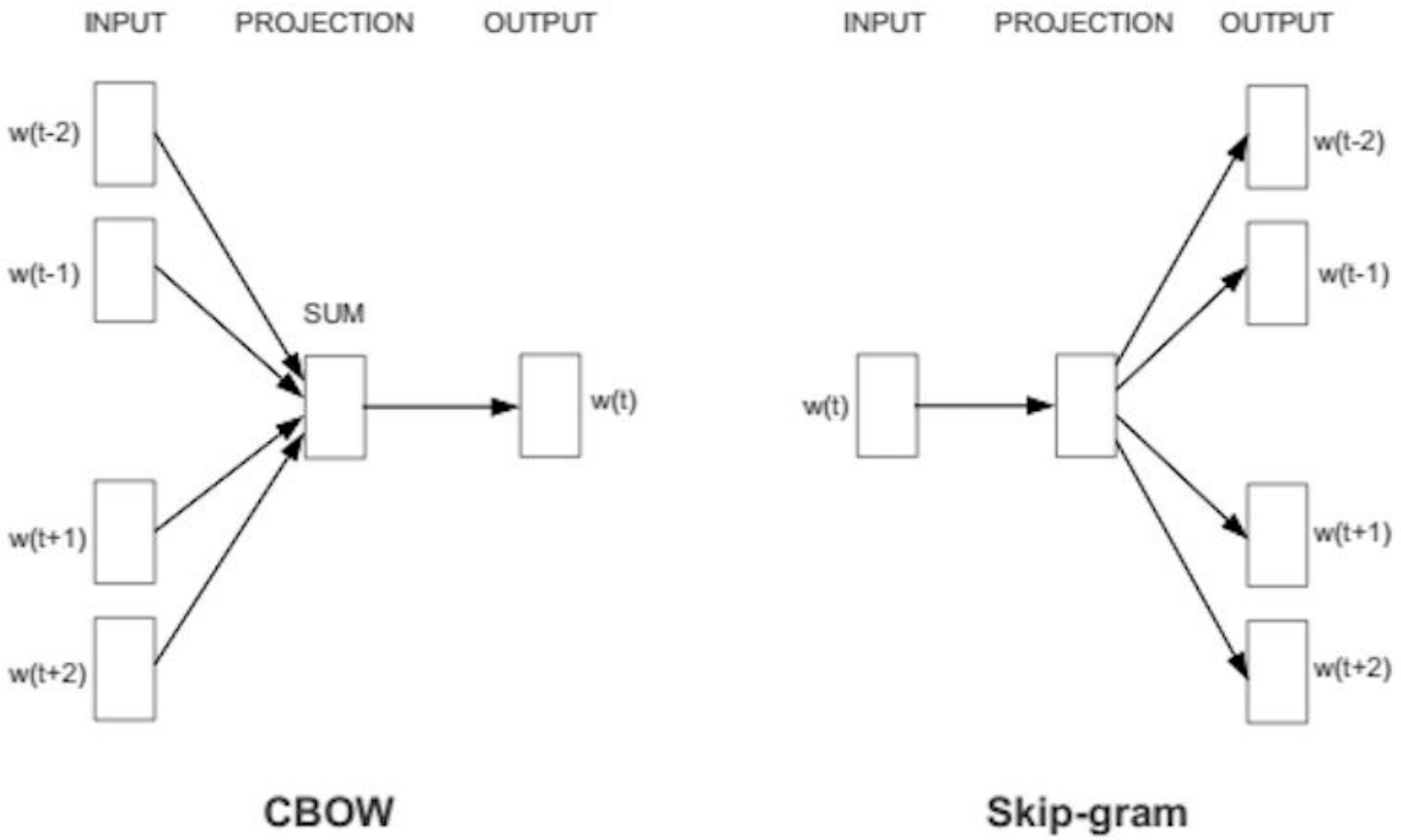
Many ways to improve results

Regression by Classification + Classification by Regression as new approach

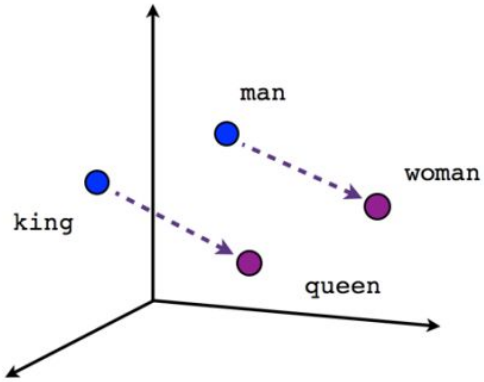
How can we identify plain text bugs?

```
void init(View decor){
    mContext = decor.getContext();
    mActionView = (ActionBarView) decor.findViewById(R.id.abs__action_bar);
    mContextView = (ActionBarContextView) decor.findViewById(R.id.abs__action_context_bar);
    mContainerView = (ActionBarContainer) decor.findViewById(R.id.abs__action_bar_container);
    mSplitView = (ActionBarContainer) decor.findViewById(R.id.abs__split_action_bar);
    if (mActionView == null || mContextView == null || mContainerView == null) {
        throw new IllegalStateException(getClass().getSimpleName() + " can only be used " + "with a compatible window decor layout");
    }
    mActionView.setContextView(mContextView);
    mContextDisplayMode = mActionView.isSplitActionBar() ? CONTEXT_DISPLAY_SPLIT : CONTEXT_DISPLAY_NORMAL;
    boolean homeButtonEnabled = mContext.getApplicationInfo().targetSdkVersion < Build.VERSION_CODES.ICE_CREAM_SANDWICH;
    homeButtonEnabled |= (mActionView.getDisplayOptions() & ActionBar.DISPLAY_HOME_AS_UP) != 0;
    setHomeButtonEnabled(homeButtonEnabled);
    setHasEmbeddedTabs(getResources_getBoolean(mContext, R.bool.abs__action_bar_embed_tabs));
}
```

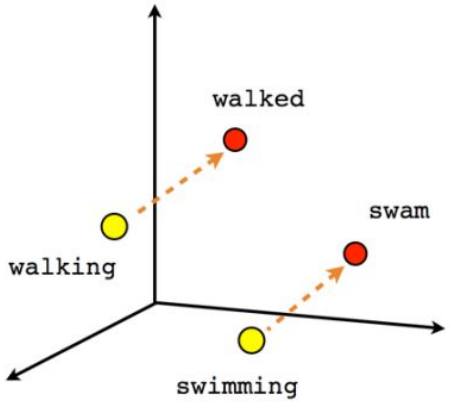
Vectorization of text (Word2Vec)



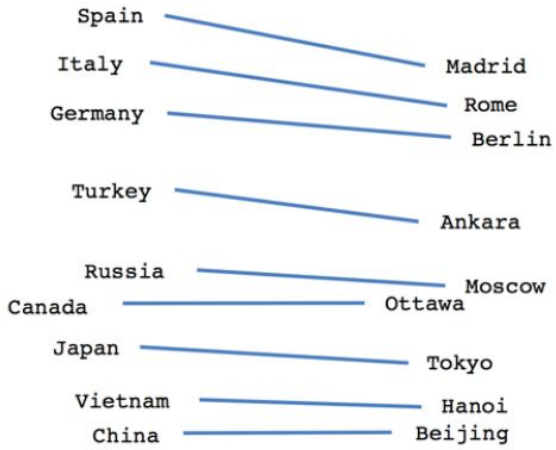
Vectorization of text



Male-Female



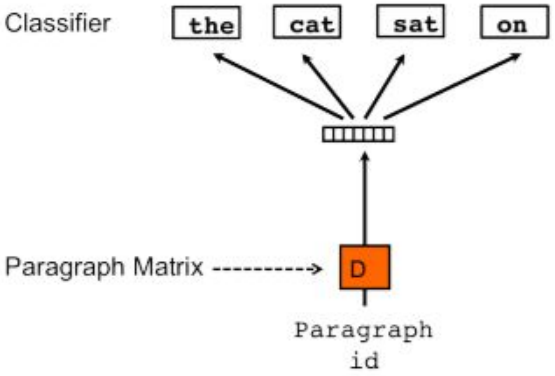
Verb tense



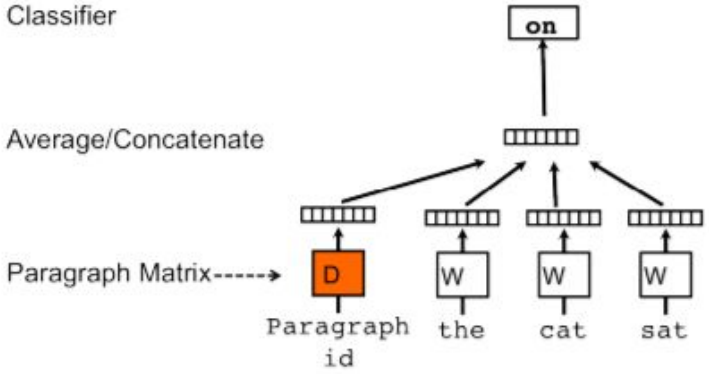
Country-Capital

Doc2Vec for classifying paragraphs

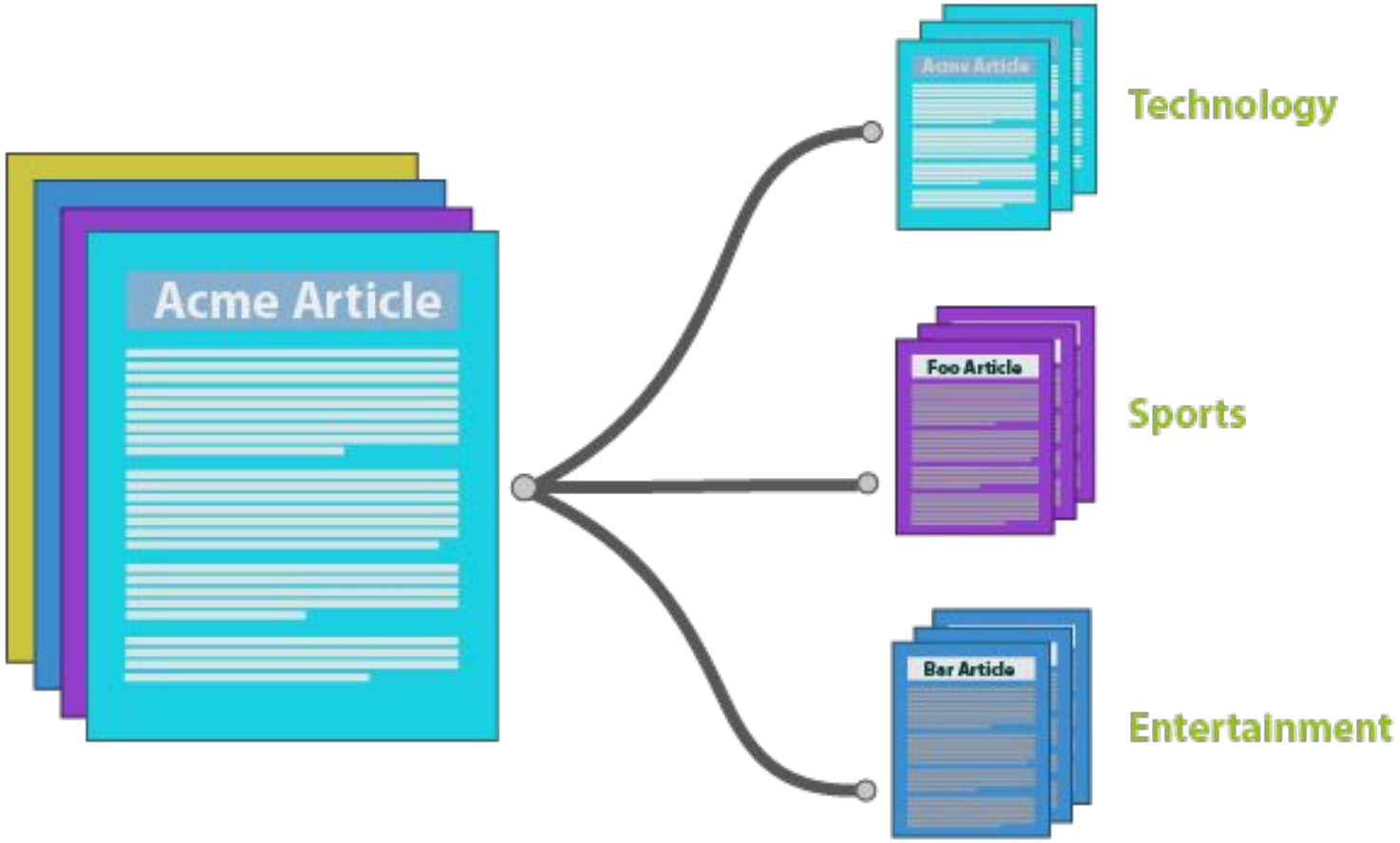
Distributed bag of words (PV-DBOW):



Distributed memory (PV-DM):



Doc2Vec for classifying paragraphs



Ground truth

Bug:

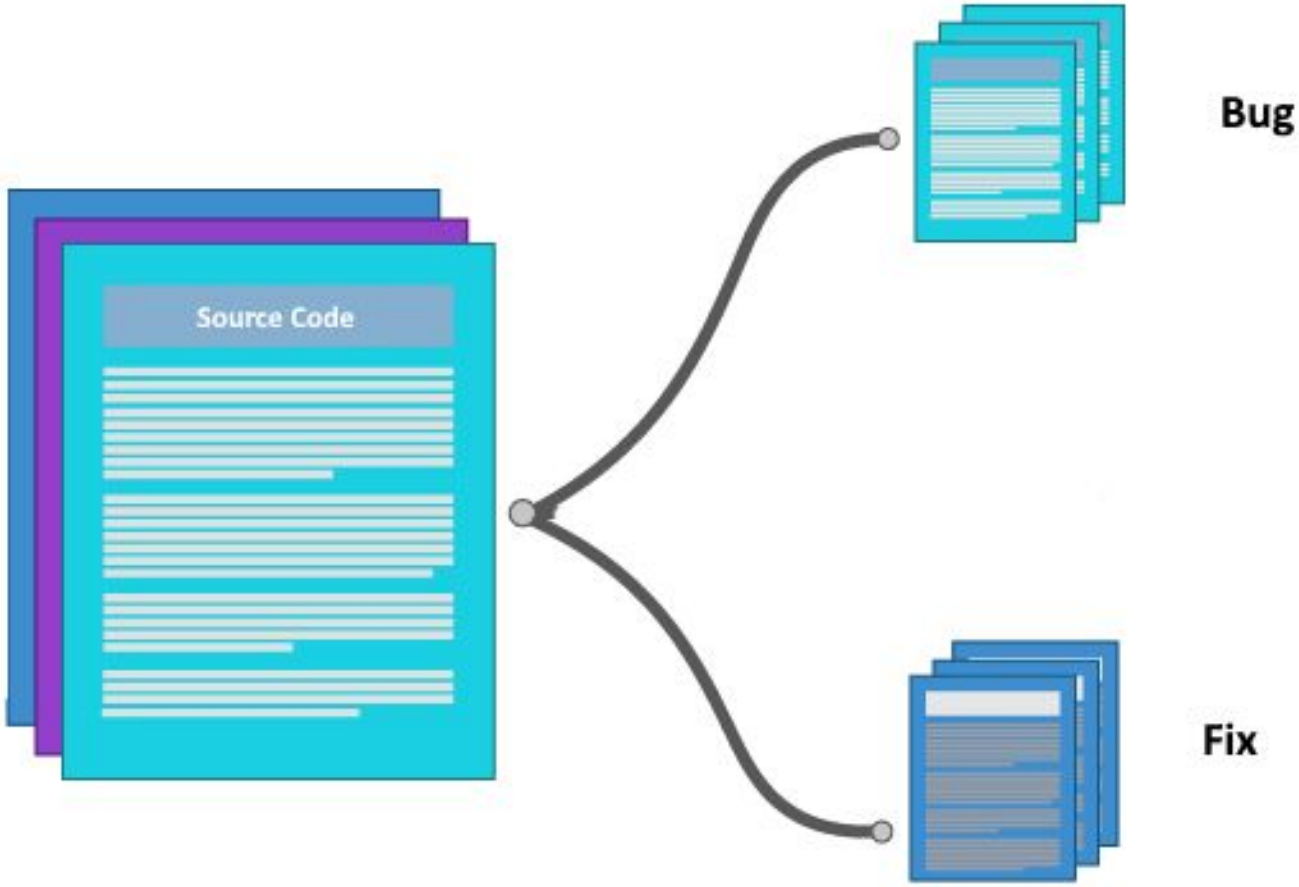
```
void init(View decor){
    mContext = decor.getContext();
    mActionView = (ActionBarView) decor.findViewById(R.id.abs__action_bar);
    mContextView = (ActionBarContextView) decor.findViewById(R.id.abs__action_context_bar);
    mContainerView = (ActionBarContainer) decor.findViewById(R.id.abs__action_bar_container);
    mSplitView = (ActionBarContainer) decor.findViewById(R.id.abs__split_action_bar);
    if (mActionView == null || mContextView == null || mContainerView == null) {
        throw new IllegalStateException(getClass().getSimpleName() + " can only be used " + "with a compatible window decor layout");
    }
    mActionView.setContextView(mContextView);
    mContextDisplayMode = mActionView.isSplitActionBar() ? CONTEXT_DISPLAY_SPLIT : CONTEXT_DISPLAY_NORMAL;
    setHomeButtonEnabled(mContext.getApplicationInfo().targetSdkVersion < 14);


    setHasEmbeddedTabs(getResources_getBoolean(mContext, R.bool.abs__action_bar_embed_tabs));
}
```

Fix:

```
void init(View decor){
    mContext = decor.getContext();
    mActionView = (ActionBarView) decor.findViewById(R.id.abs__action_bar);
    mContextView = (ActionBarContextView) decor.findViewById(R.id.abs__action_context_bar);
    mContainerView = (ActionBarContainer) decor.findViewById(R.id.abs__action_bar_container);
    mSplitView = (ActionBarContainer) decor.findViewById(R.id.abs__split_action_bar);
    if (mActionView == null || mContextView == null || mContainerView == null) {
        throw new IllegalStateException(getClass().getSimpleName() + " can only be used " + "with a compatible window decor layout");
    }
    mActionView.setContextView(mContextView);
    mContextDisplayMode = mActionView.isSplitActionBar() ? CONTEXT_DISPLAY_SPLIT : CONTEXT_DISPLAY_NORMAL;
    boolean homeButtonEnabled = mContext.getApplicationInfo().targetSdkVersion < Build.VERSION_CODES.ICE_CREAM_SANDWICH;
    homeButtonEnabled |= (mActionView.getDisplayOptions() & ActionBar.DISPLAY_HOME_AS_UP) != 0;
    setHomeButtonEnabled(homeButtonEnabled);
    setHasEmbeddedTabs(getResources_getBoolean(mContext, R.bool.abs__action_bar_embed_tabs));
}
```

Goal



Results PV-DM

Experiment	Accuracy	Precision	Recall
Raw data	0.5	0.24	0.34
Anonymization	0.5	0	0
Scoping with anonymization	0.5	0	0
Scoping without anonymization	0.5	0.25	0.38

Results PV-DBOW

Experiment	Accuracy	Precision	Recall
Raw data	0.5	0.25	0.12
Anonymization	0.5	0	0
Scoping with anonymization	0.5	0	0
Scoping without anonymization	0.5	0.25	0.53

Conclusion

- Code should not be treated as simple text

- More information and more complex model needed for successful bug prediction

Different approach - code metrics

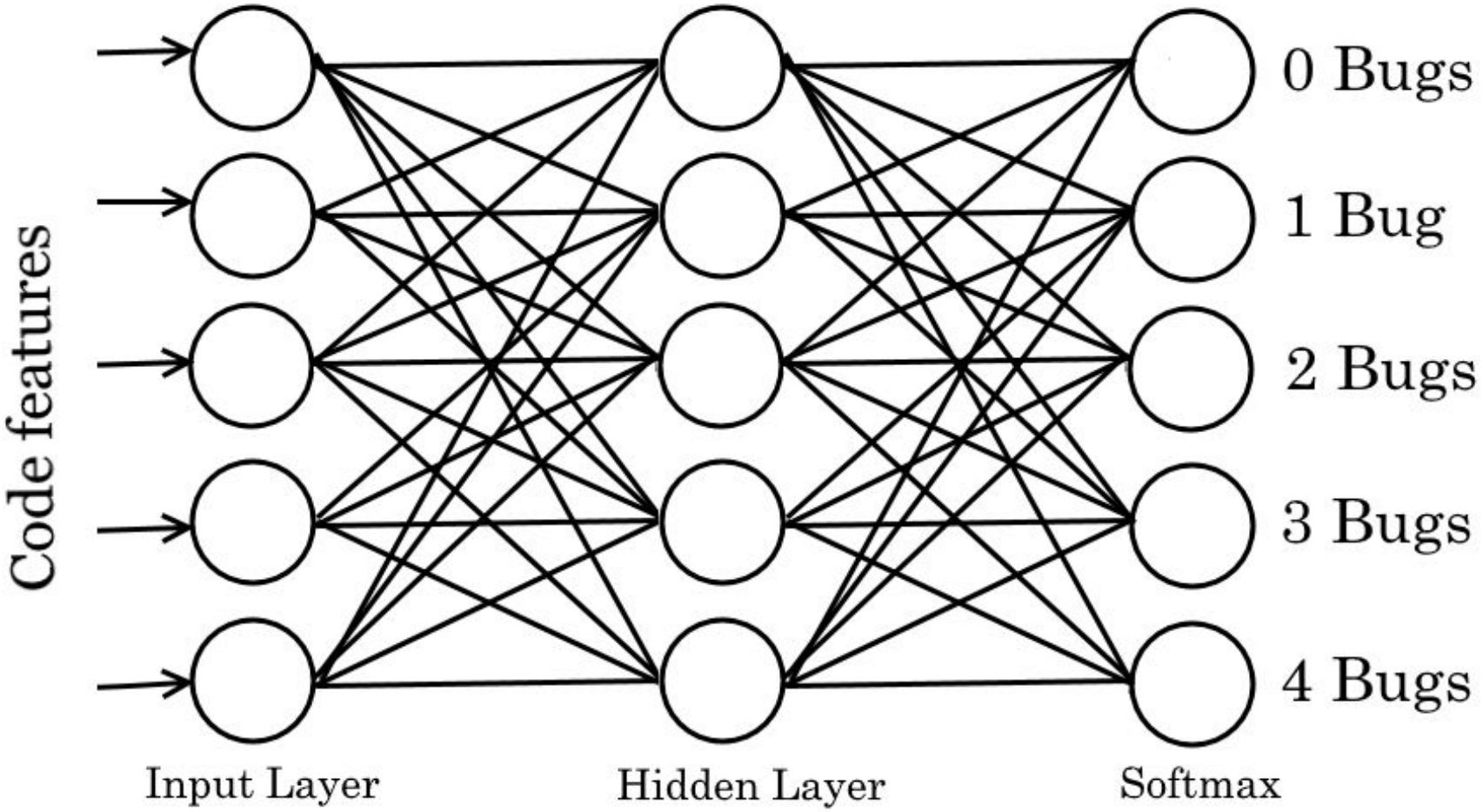
Software metric sets of 14 different projects

20 - 32 features

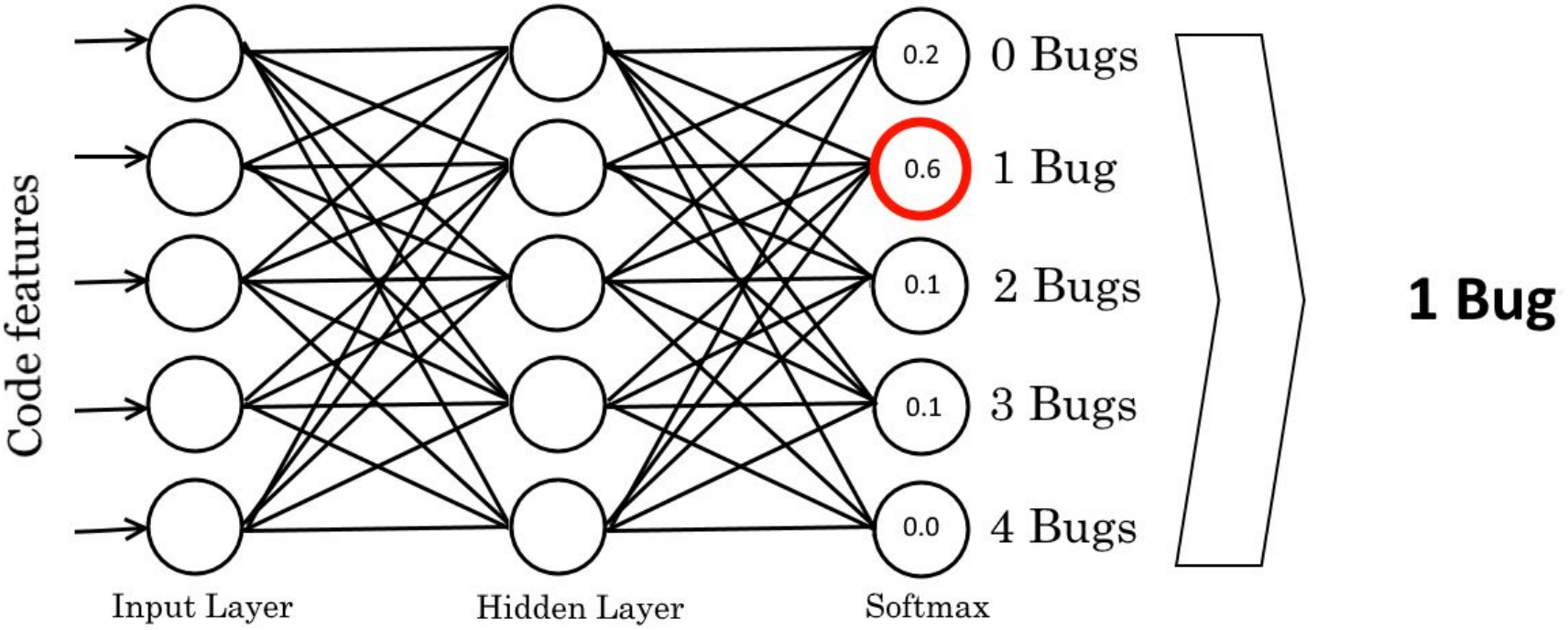
Defect count as response variable

name	loc	...	max_cc	avg_cc	bug
SAXXMLOutput	509	...	1	0.8125	2

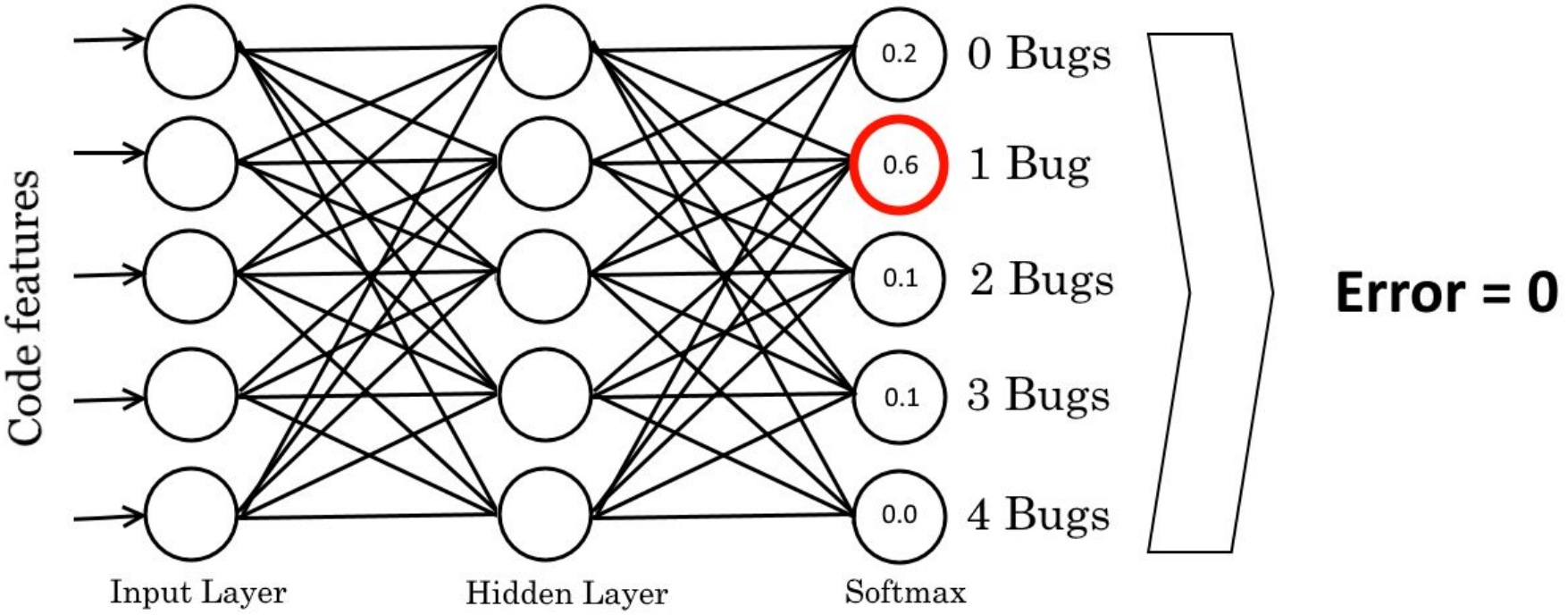
Feedforward neural net



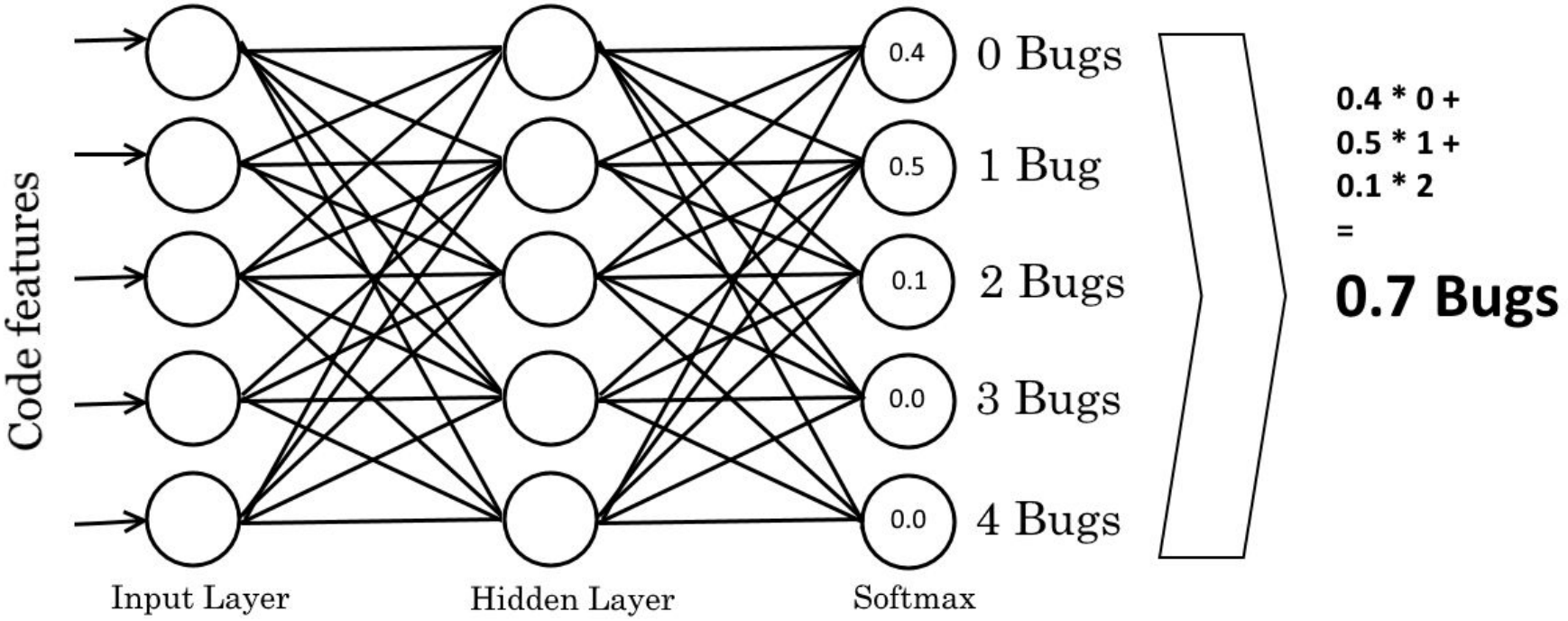
Classification



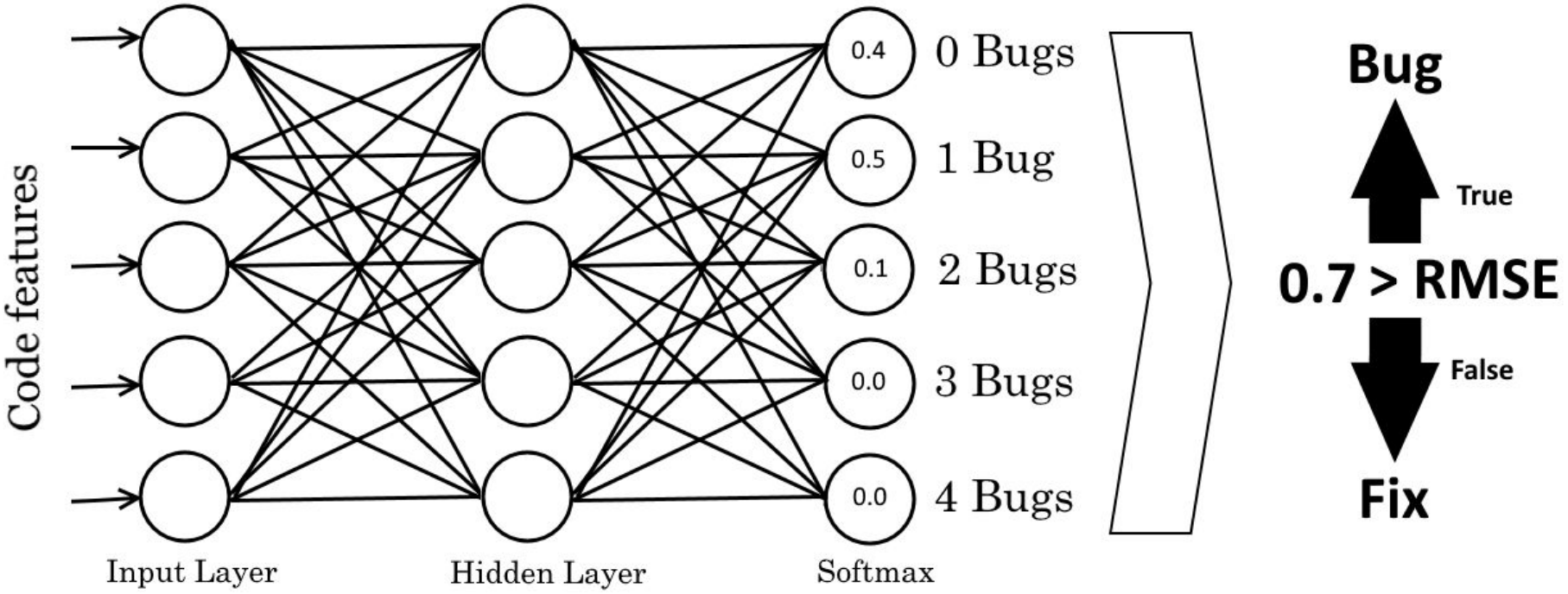
Regression by Classification (RbC)



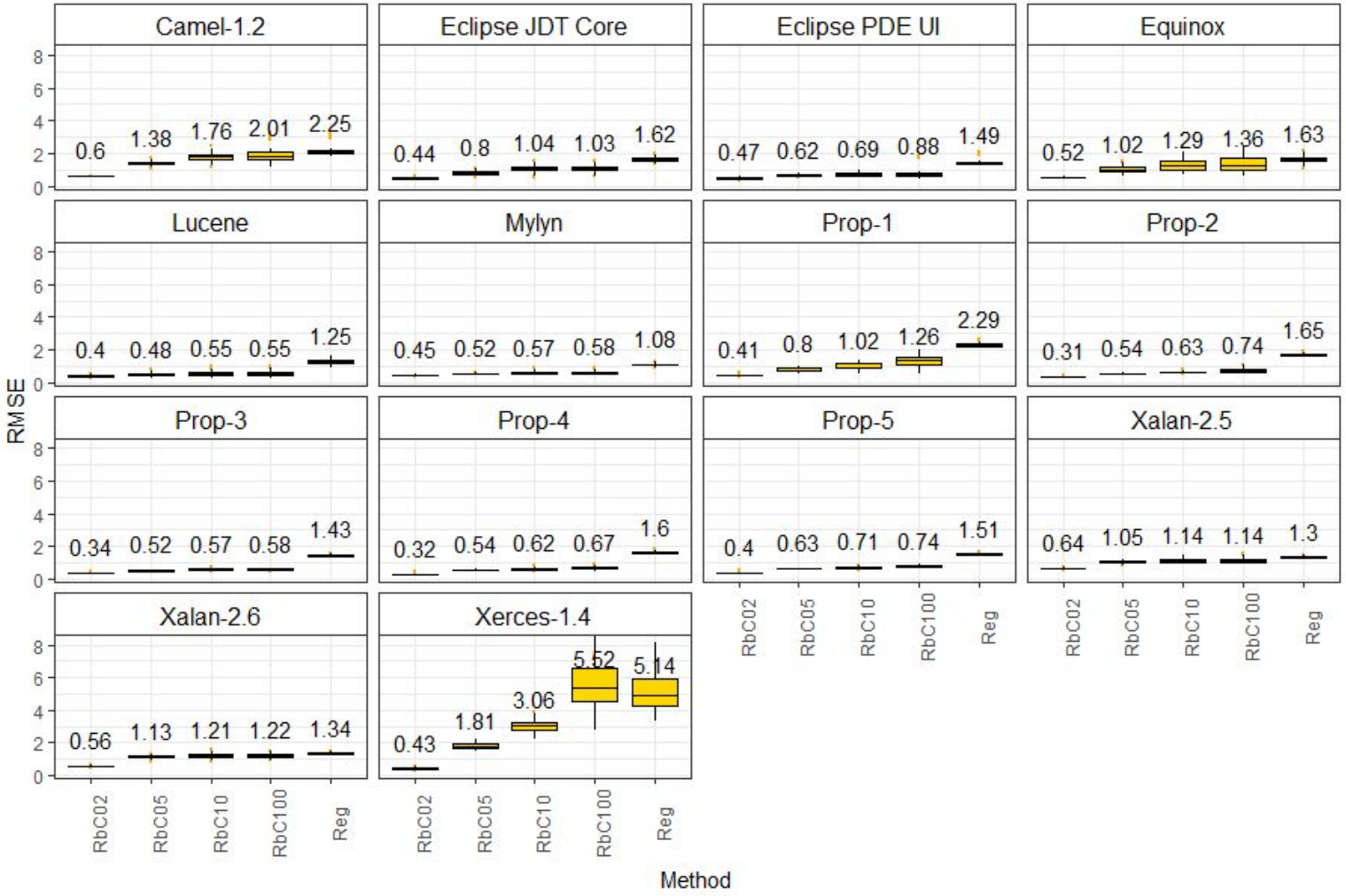
Regression


Code features
Input Layer
Hidden Layer
Softmax
0.4
0.5
0.1
0.0
0.0
0 Bugs
1 Bug
2 Bugs
3 Bugs
4 Bugs
0.4 * 0 +
0.5 * 1 +
0.1 * 2
=
0.7 Bugs

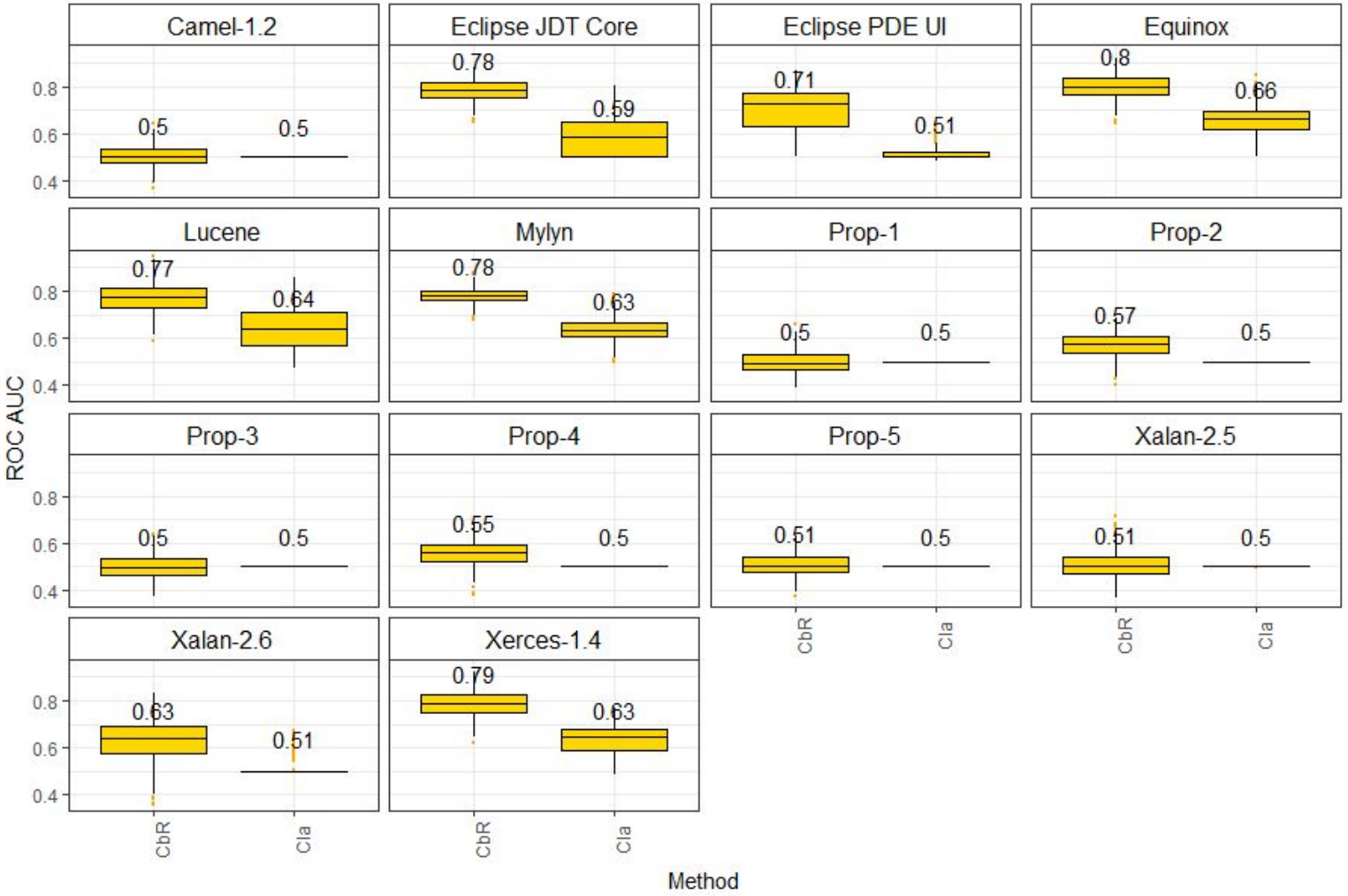
Classification by Regression (CbR)



Results regression vs RbC



Results classification vs CbR



Conclusion

- CbR and RbC can increase the performance of standard regression and classification in bug prediction

- Better results can be achieved with some more tweaking

Future work

- Hyperparameter optimization, feature selection
- LSTM approach
- “Code2Vec” with additional input about context