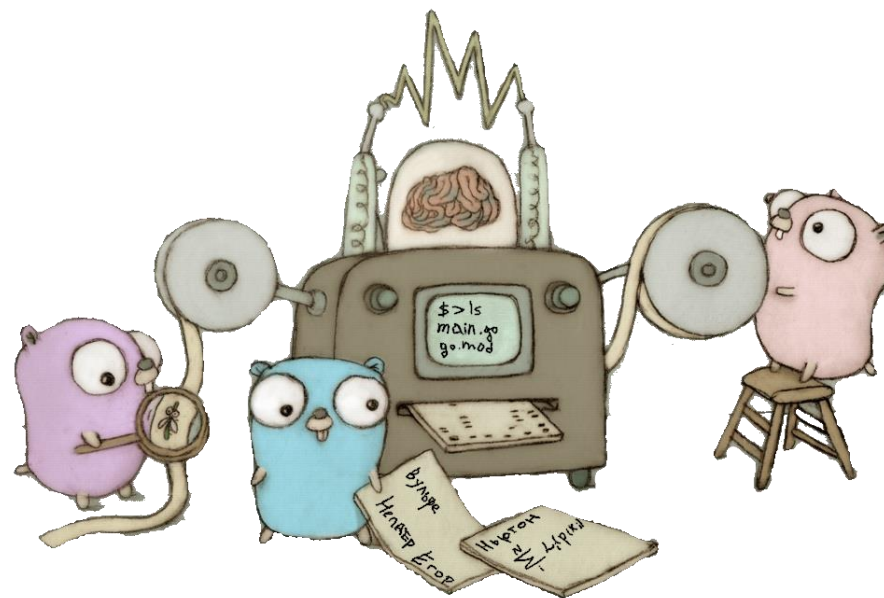


# Наша Гордость

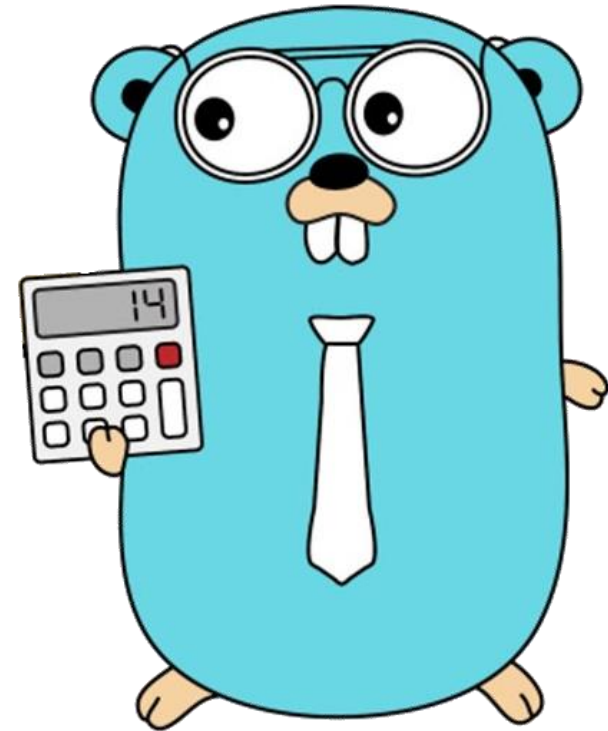
Линейная регрессия на Go  
против питона

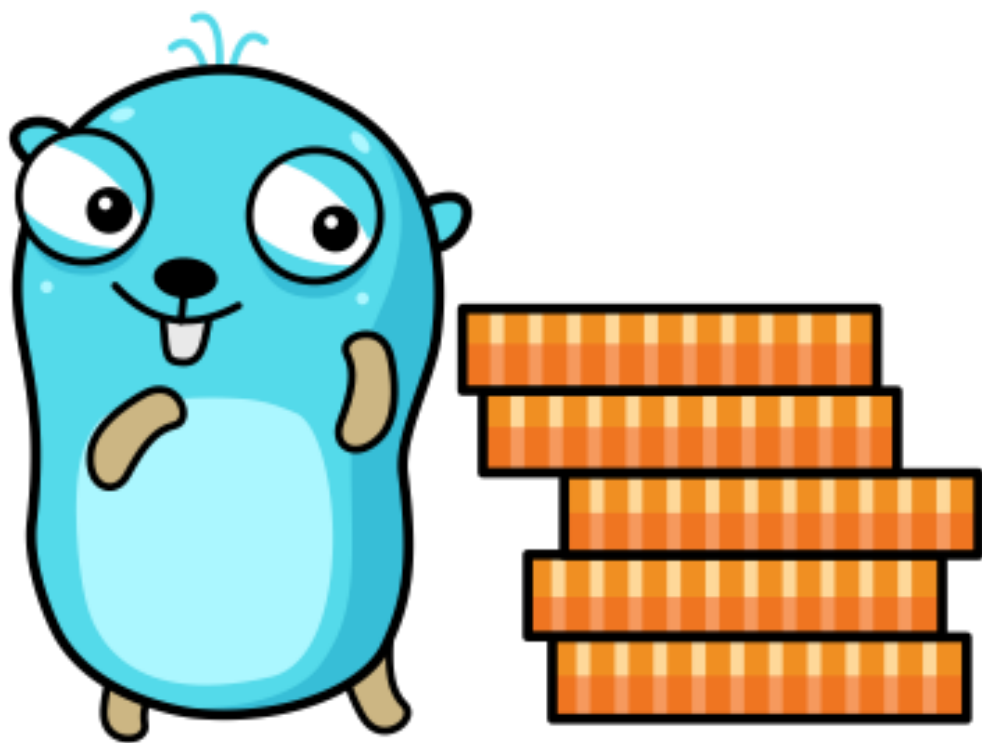
Гуров Матвей  
Тарасов Иван  
Фараонов Егор



# Почему Go?

- Быстрый
- Типизируемый
- Параллельный
- Мы его любим
- Гофер





# План рассказа

- Немного про код
- Что он умеет?
- Сравнение с питоном

# Наш код



```
type Loss interface {
    F(yPred, y float64) float64

    // Df is d(F)/d(Model.Predict)
    Df(yPred, y float64) float64
}
```

```
type Row struct {
    X mat.Vector
    Y float64
}

type DataSet interface {
    Row(idx int) Row
    Len() int
    Dim() int
}
```

```
type Trainer interface {
    Train(ml.Model, DataSet)
}
```

```
type Model interface {
    Predict(x mat.Vector) float64
    Config() Config
    Weights() *mat.VecDense
    SetWeights(mat.Vector)
    Bias() float64
    SetBias(float64)

    // Dp is d(Predict)/d(Weights)
    Dp(x mat.Vector) mat.Vector
}

type Config struct {
    // RowLen is inputs len
    RowLen int
    Loss    Loss
    Reg     Regularizator
    Bias    bool
}
```

```
data, _ := training.NewSliceDatasetFromCSV("test.csv")

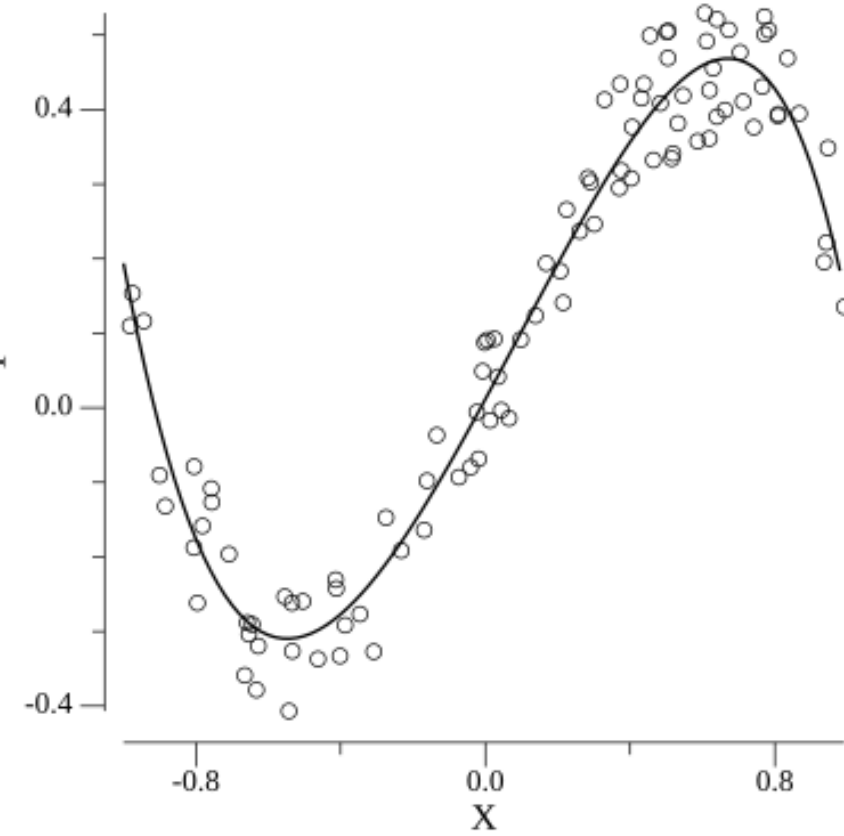
model := ml.NewLinearModel(ml.Config{
    RowLen: 2,
    Loss:   ml.MSELoss{},
    Reg:    ml.EmptyRegularizator{},
    Bias:   true,
})

trainer := training.NewBatchTrainer(
    1000,
    10000,
    training.GeometricLearningRate(1, 0.99999),
)

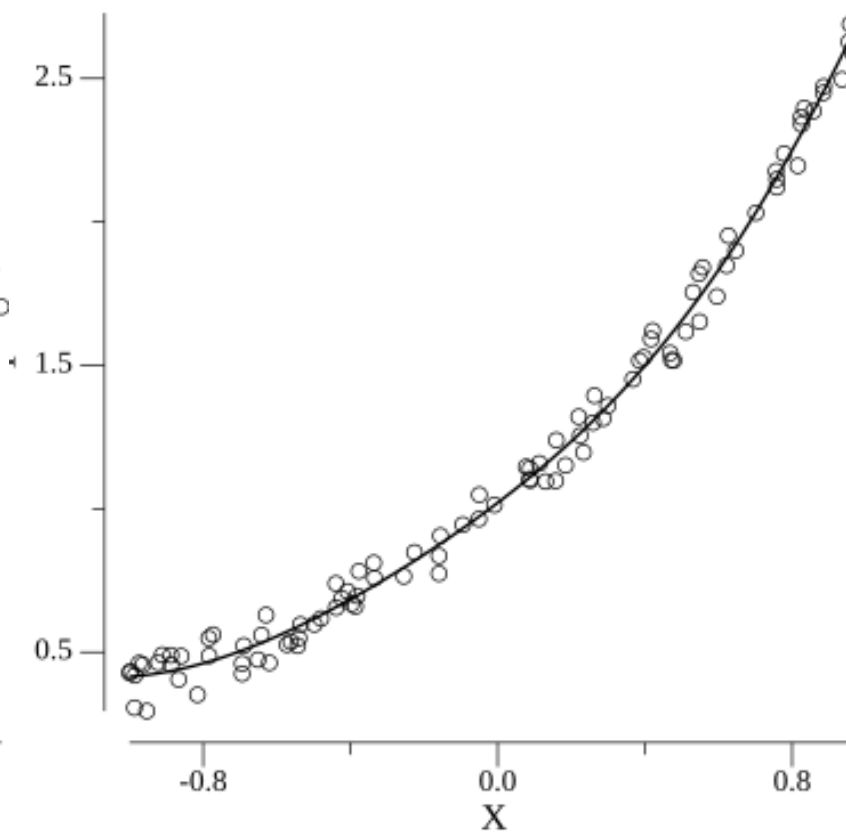
trainer.Train(model, data)

fmt.Println(
    model.Predict(mat.NewVecDense(2, []float64{1, 2})),
)
```

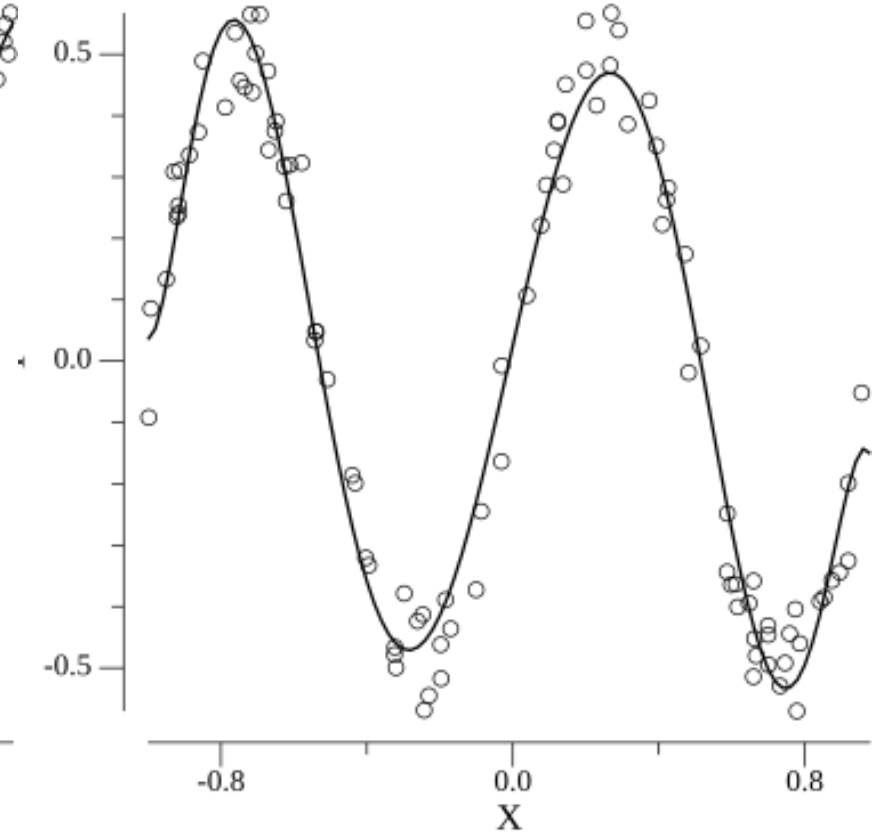
DataSet Plot



DataSet Plot



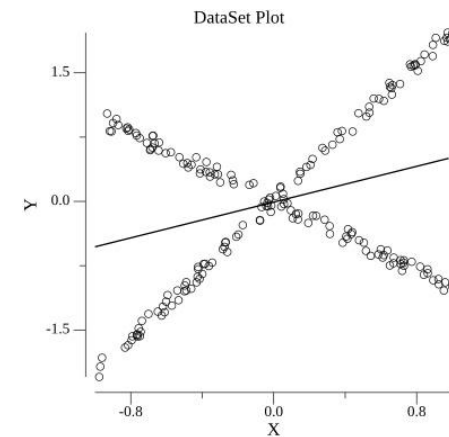
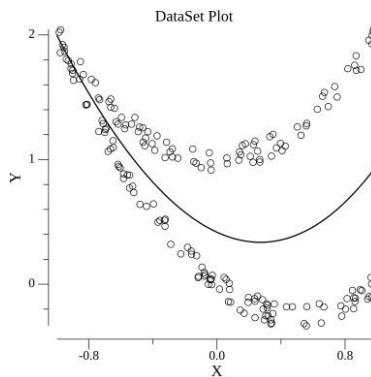
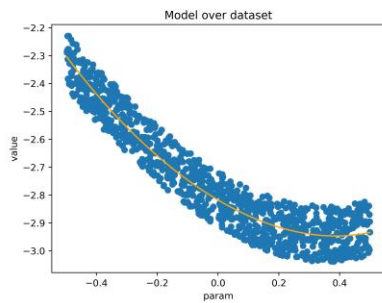
DataSet Plot



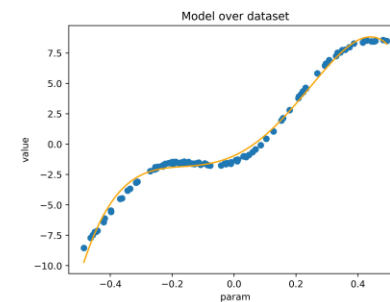
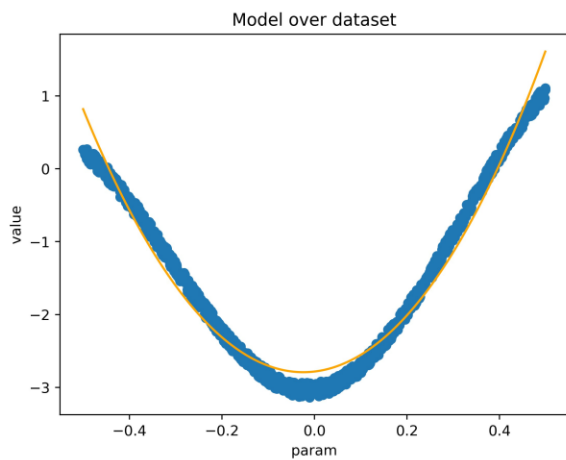
---

Аппроксимация полиномом





# Аппроксимация ПОЛИНОМОМ



# Сравнение с ПИТОНОМ



Dim: 100 | Len: 5000

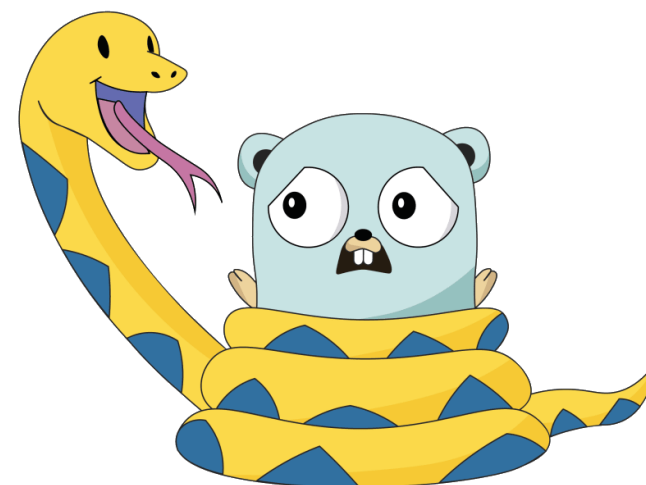
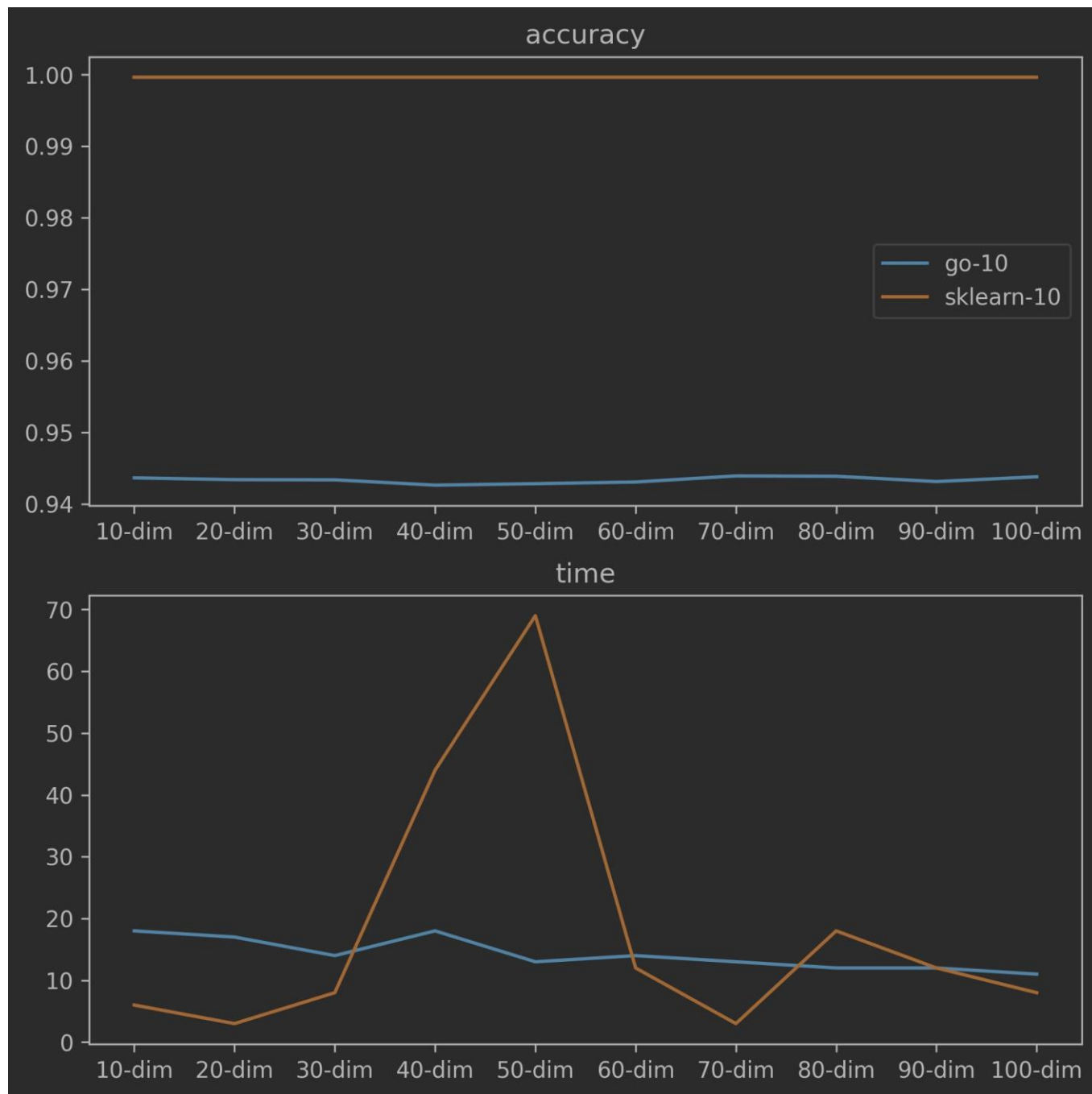
name	accuracy	time
Batch	0.99976985	261
Sklearn	0.99984805	433

Dim: 200 | Len: 2000

name	accuracy	time
Batch	0.99948801	175
Sklearn	0.99990301	742



# Сравнение с ПИТОНОМ



Наш репозиторий



Наша команда

