Учреждение образования

«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра информатики

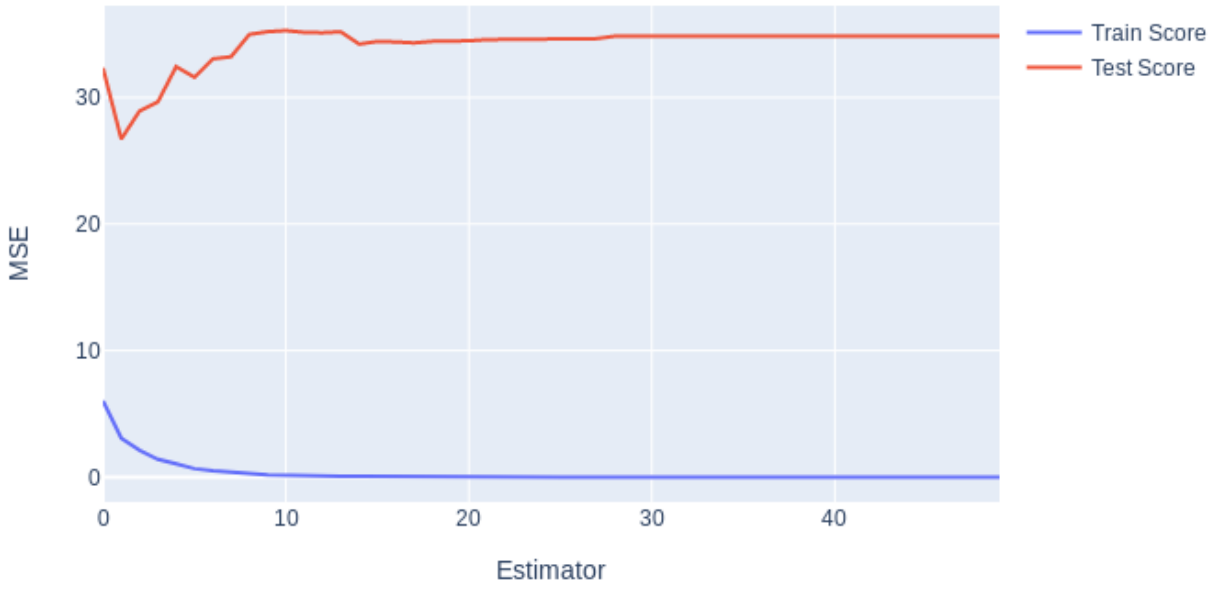
Тема отчёта:
«Градиентный бустинг»

Выполнил: Демидов Дмитрий Александрович
магистрант кафедры информатики
группа №858642

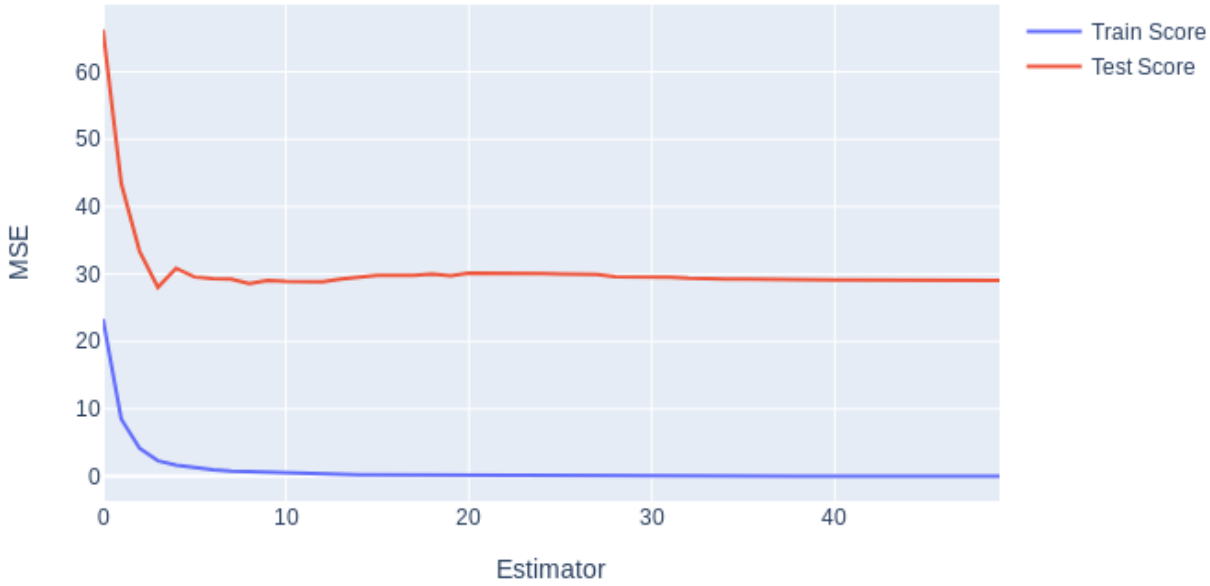
Проверил: магистр технических наук
Стержанов Максим Валерьевич

Минск 2019

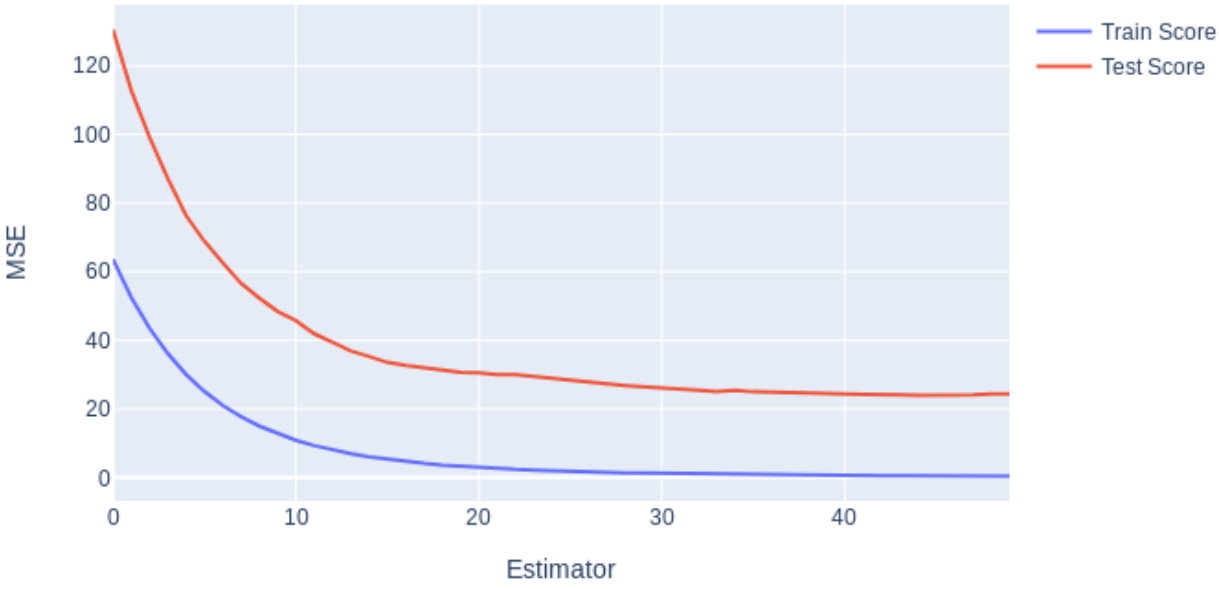
Исследуйте, переобучается ли градиентный бустинг с ростом числа итераций, а также с ростом глубины деревьев. Постройте графики.



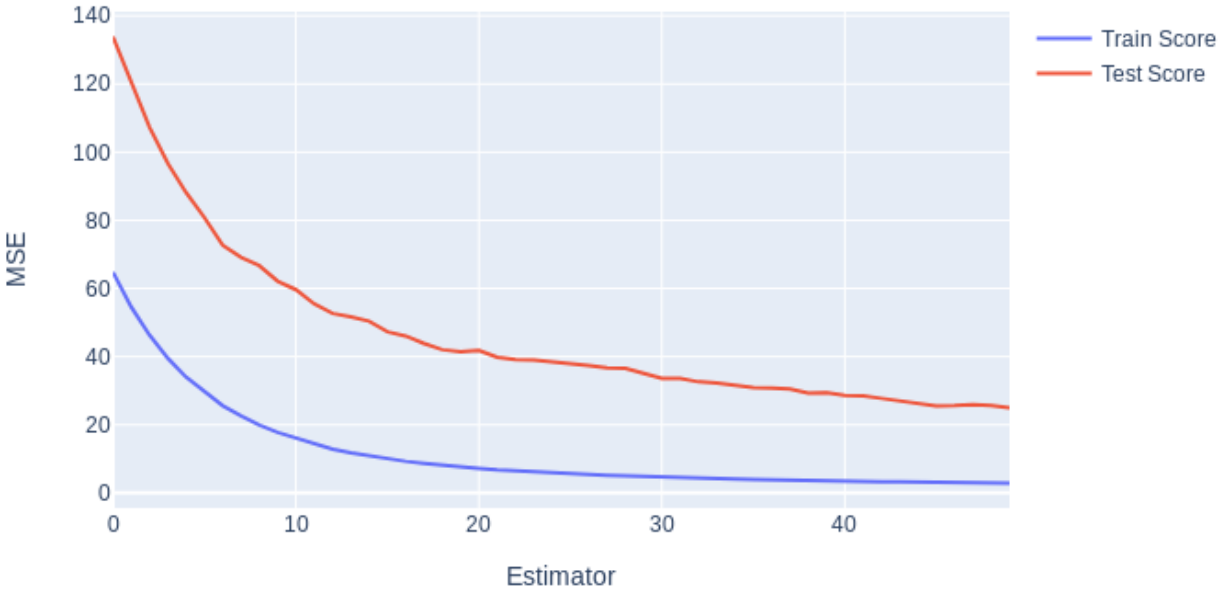
max_depth=5, n_estimators=50, ***learning_rate=0.9***



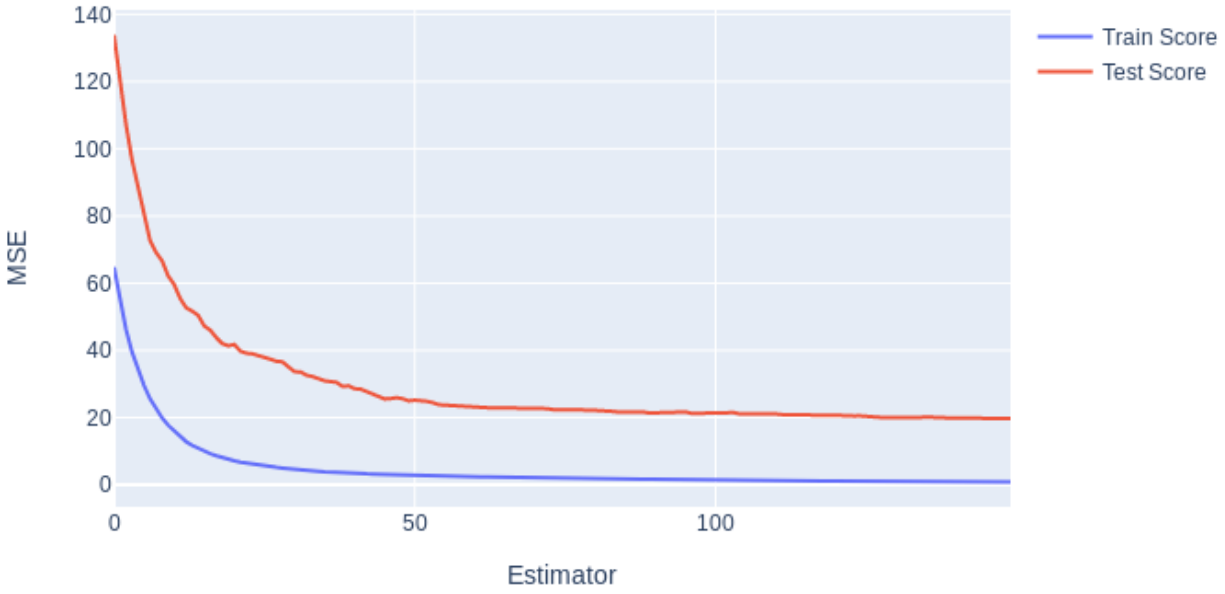
learning_rate=0.5



learning_rate=0.1 (MSE=24)



***max_depth=3**, n_estimators=50, learning_rate=0.1 (MSE=25)*



n_estimators=150 (MSE=20)

*Уменьшение **learning_rate** повышает точность предсказания, но может понадоится увеличение количества используемых деревьев.*

*Увелисение **максимальной глубины** деревьев уменьшает необходимое количество деревьев, но изменение глубины в любую сторону от оптимальной ведёт к понижению точности.*

*Увеличение **количества деревьев** не ведёт к переобучению.*

Сравните качество, получаемое с помощью градиентного бустинга с качеством работы линейной регрессии.

```
from sklearn.linear_model import LinearRegression, Ridge
from sklearn.metrics import mean_squared_error as mse

model = LinearRegression()
model.fit(X_train, y_train)
print('Linear MSE: %s' % mse(y_test, model.predict(X_test)))

model = Ridge(alpha=200)
model.fit(X_train, y_train)
print('Ridge MSE: %s' % mse(y_test, model.predict(X_test)))
```

```
Linear MSE: 78.99626253782839
Ridge MSE: 50.717624018293904
```

На тестовой выборке Градиентный бустинг показал себя значительно лучше линейных моделей.

Но на контрольной выборке **линейная регрессия показала себя луче** всех остальных двух.

```
final_train_size = train_size+test_size
X_final_train = X[:final_train_size]
y_final_train = y[:final_train_size]
X_holdout = X[final_train_size:]
y_holdout = y[final_train_size:]

model = GradientBoostingRegressor(max_depth=3, random_state=42, n_estimators=150, learning_rate=0.1)
model.fit(X_final_train, y_final_train)
print('GBR MSE: %s' % mse(y_holdout, model.predict(X_holdout)))

model = LinearRegression()
model.fit(X_final_train, y_final_train)
print('Linear MSE: %s' % mse(y_holdout, model.predict(X_holdout)))

model = Ridge(alpha=200)
model.fit(X_final_train, y_final_train)
print('Ridge MSE: %s' % mse(y_holdout, model.predict(X_holdout)))
```

```
GBR MSE: 16.562460165161575
Linear MSE: 14.913375625428985
Ridge MSE: 20.77715153066596
```

Увеличение количества элементов в обучающей выборке ведёт к необходимости увеличения количества деревьев и уменьшения их глубины. В данном примере, чтобы **сравняться** по точности с линейной регрессией, необходимо было уменшить **максимальную глубину до 2**. Чтобы **превзойти по точности** (13 против 15 у линейной регрессии), необходимо было **увеличить количество деревьев до 250**.

```
model = GradientBoostingRegressor(max_depth=2, random_state=42, n_estimators=250, learning_rate=0.1)
model.fit(X_final_train, y_final_train)
print('GBR MSE: %s' % mse(y_holdout, model.predict(X_holdout)))
```

```
GBR MSE: 12.918745233167787
```