# Презентация по курсовому проекту

# Модель которая используется

В данном курсовом использовалась модель машинного обучения классификатор LGBMClassifier. Соответственно перед тем как обучать модель классификации была проведена стандартизация признаков, так как все признаки были разного рода показателей.

# Сравнение моделей

Для сравнения были использованы модели:

1) RandomForestClassifier
2) CatBoostClassifier
3) LGBMClassifier

На мой взгляд эти модели чаще всего используются в классификации, поэтому и я решил использовать эти модели

# Каждая модель была протестирована через кросс-валидацию и были получены следующие результаты

```
rf = RandomForestClassifier(random_state=42, n_jobs=-1, max_depth=4)
cv_score = cross_val_score(
    rf,
    X_train,
    y_train,
    scoring='f1',
    cv=StratifiedKFold(n_splits=5, shuffle=True, random_state=21)
)
cv_score
# rf.fit(X_train, y_train)

# y_train_pred = rf.predict(X_train)
# y_test_pred = rf.predict(X_test)



# get_classification_report(y_train, y_train_pred, y_test, y_test_pred)
```

```
array([0.74115086, 0.74306662, 0.74158122, 0.7427757 , 0.74422364])
```

```
catb = CatBoostClassifier(silent=True,learning_rate=0.01, max_depth=4,iterations=50, random_state=21)

cv_score = cross_val_score(
    catb,
    X_train,
    y_train,
    scoring='f1',
    cv=StratifiedKFold(n_splits=5, shuffle=True, random_state=21)
)
cv_score
# catb.fit(X_train, y_train)

# y_train_pred = catb.predict(X_train)
# y_test_pred = catb.predict(X_test)

# get_classification_report(y_train, y_train_pred, y_test, y_test_pred)
```

```
array([0.87815264, 0.87823924, 0.87755741, 0.87779069, 0.87830867])
```

```
model_lgbm = LGBMClassifier(random_state=21, n_estimators=100)
cv_score = cross_val_score(
    model_lgbm,
    X_train,
    y_train,
    scoring='f1',
    cv=StratifiedKFold(n_splits=5, shuffle=True, random_state=21)
)
cv_score
# model_lgbm.fit(X_train, y_train)

# y_train_pred = model_lgbm.predict(X_train)
# y_test_pred = model_lgbm.predict(X_test)


# get_classification_report(y_train, y_train_pred, y_test, y_test_pred)
```

```
array([0.88689903, 0.88704687, 0.88681679, 0.88615292, 0.88711264])
```

# Принцип составления предложений

Многие признаки были не важными, более 90% значений этих признаков принимали только одно значение, естественно такие признаки не показатели и от них я избавился.

Так же в модели явное преимущество target это 0, из-за чего модель будет плохо обучаться, для этого использовал баланс target добавляя новые строки что бы соотношение target 0 и target 1 было практически одинаковым.