Курсовая

Курсовая работа на тему: "Построение моделей машинного обучения для оценки эффективности химических соединений против вируса гриппа"

Параметры оценки противовирусной активности: IC50, CC50, SI

Ключевые показатели эффективности

В исследованиях противовирусных соединений, особенно против вируса гриппа, используются три фундаментальных параметра:

1. **IC50 (Half Maximal Inhibitory Concentration)**
   - Концентрация, необходимая для подавления вирусной репликации на 50%
   - **Интерпретация**: Чем **ниже** значение, тем выше противовирусная активность
   - Единицы: μM или nM
   - *Пример*: Соединение с IC50=0.1 μM эффективнее, чем с IC50=10 μM
2. **CC50 (Half Maximal Cytotoxic Concentration)**
   - Концентрация, вызывающая гибель 50% здоровых клеток
   - **Интерпретация**: Чем **выше** значение, тем ниже токсичность
   - Единицы: μM или nM (согласованы с IC50)
   - *Пример*: Соединение с CC50=100 μM безопаснее, чем с CC50=5 μM
3. **SI (Selectivity Index)**
   - Расчетный параметр: **SI = CC50 / IC50**
   - **Интерпретация**:
     - Показатель терапевтического окна
     - Чем **выше** значение, тем лучше профиль безопасности
     - Минимальный порог для перспективных соединений: SI > 8
   - *Пример*: При CC50=100 μM и IC50=1 μM → SI=100 (отличный показатель)

О техническом описании представленных к анализу соединений и химической информатике

Остальные представленные в датафрейме признаки являются молекулярными дискрипоторами соединений , извлеченными возможно с помощью библиотеки RDKit. RDKit - набор инструментов с открытым исходным кодом для химической информатики. Он был разработан Грегом Лэндрамом с многочисленными дополнениями от сообщества разработчиков RDKit с открытым исходным кодом. Он имеет интерфейс прикладного программирования (API) для Python, Java, C++ и C#. Позволяет раскрывать признаковое описание каждой молекулы, зашифрованной в разные форматы, например: Smiles. Кроме формата Smiles, научное сообщество использует .sdf, SMILES (InChi, SMARTS, SYBYL, SELFIES), .cif (твердые тела), .pdb(белки и нуклеиновые кислоты)

Базы данных:

1. PubChem – база данных органических соединений, с огромным объемом. K

сожалению, в ней существуют повторные записи, недостающие данные и т.д. На текущий момент в ней 110 млн. соединений. 2. ChEMBL – база данных, которая ориентируется на биологически активные соединения. Кроме того включает большое количество свойств помимо целевых активностей. 3. ZINC53 – интересна тем, что основной мотив для включения в базу данных – коммерческая доступность. 4. PDB – база данных для структур белков. Существуют уже разрешенные комплексы белок/лиганд. 5. ICSD – база данные неорганических соединений, структуры 6. CSD – база данных просто кристаллических структур 7. Crystallography Open Database – структуры, можно полностью сказать.

Молекулярные дискрипторы бывают в виде: 0D, 1D, 2D, 3D, 4D. Описываю многомерные структуры молекул. В зависимости от задачи. Так например: 3D/4D-дескрипторы используют информацию о 3D структуре молекулы, а 4D о конформациях.

Химическая информатика наработала колосальный объем информации об элементарных частицах, молекулах и соединениях, а также накопила достаточно данных о воздействии этих веществ на объекты материи в т.ч. живые организмы, что позволило интегрировать накопленные знания в различные типы моделей машинного обучения для поиска неообходимых решений.

Например:

- классификации соединений по различным основаниям
- предсказания характеристик, степени воздействия на организмы и др.
- создания новых соединений и структур

В данной работе мы рассмотрим различные типы моделей машинного обучения (способ обучения - с учителем, т.к. у нас есть конкретные таргеты) и выберем наиболее подходящие для решения нашей основной задачи: "Оценки эффективности химических соединений против вируса гриппа по признаковому описанию (молекулярных дексрипторов)".

Классификация молекулярных дескрипторов

Фармакологические параметры

- `IC50, mM` : Концентрация для 50% подавления вируса
- `CC50, mM` : Концентрация для 50% гибели клеток
- `SI` : Индекс селективности (CC50/IC50)

Электронные состояния (EState)

- **Атомные индексы**:
  `MaxAbsEStateIndex` , `MaxEStateIndex` ,
  `MinAbsEStateIndex` , `MinEStateIndex`
- **Поверхностные дескрипторы**:
  `EState_VSA1-11` , `VSA_EState1-10`

Количественно описывают распределение электронной плотности в молекуле:

- Атомные индексы (Max/Min EStateIndex): Характеризуют реакционную способность отдельных атомов
- VSA-дескрипторы (EState_VSA, VSA_EState): Связывают электронные свойства с площадью поверхности

MaxEStateIndex: Атом с наивысшей электронной плотностью (потенциальный сайт реакций)

EState_VSA5: Вклад атомов со средними значениями EState в площадь поверхности

Физико-химические свойства

Группа	Дескрипторы	Значимость
Масса/состав	`MolWt` , `HeavyAtomMolWt` , `ExactMolWt` , `HeavyAtomCount` , `NumHeteroatoms`	Влияет на проникновение через мембраны
Липофильность	`MolLogP` , `BCUT2D_LOGPHI` , `BCUT2D_LOGPLOW`	Определяет распределение в организме
Полярность	`TPSA` , `LabuteASA`	Влияет на растворимость и связывание
Рефракция/ заряд	`MolMR` , `BCUT2D_MRHI` , `BCUT2D_MRLOW` , `MaxPartialCharge` , `MinPartialCharge`	Определяет электростатические взаимодействия

Топологические индексы

- **Сложность структуры**:
  `BalabanJ` , `BertzCT` , `Kappa1-3` , `Ipc` , `AvgIpc` - (BertzCT): Отражает "ветвистость" молекулы
- **Связность**:
  `Chi0-4n` , `Chi0-4v` , `HallKierAlpha` - Описывают пути через атомы

- **Гибкость/циклы**:
  `NumRotatableBonds` , `RingCount` , `NumAliphaticRings` , `NumAromaticRings` NumRotatableBonds - Влияет на конформационную энтропию

Поверхностные дескрипторы (VSA)

- **По заряду**: `PEOE_VSA1-14` - Распределение парциальных зарядов по поверхности
- **По молекулярной рефракции**: `SMR_VSA1-10` - Вклад в молекулярную рефракцию
- **По липофильности**: `SlogP_VSA1-12` - Гидрофобные/гидрофильные участки поверхности

Электронные параметры

- **Электронная структура**:
  `NumValenceElectrons` , `NumRadicalElectrons` ,
  `BCUT2D_MWHI` , `BCUT2D_MWLOW` , `BCUT2D_CHGHI` , `BCUT2D_CHGLO`
- BCUT2D-дескрипторы: Интегрируют массу, заряд, поляризуемость
- MWHI/MWLOW: Весовые коэффициенты для тяжелых атомов
- CHGHI/CHGLO: Экстремумы зарядового распределения
- **Фингерпринты**:
  `FpDensityMorgan1-3` - Локальное атомное окружение

Функциональные группы (fr_*):

- ['fr_COO', 'fr_Ar_OH', 'fr_Al_OH'] # Кислоты/фенолы - Ионизация, растворимость
- ['fr_aldehyde', 'fr_ketone', 'fr_ester'] # Карбонилы - Водородные связи, реакционность
- ['fr_halogen', 'fr_nitro', 'fr_ether'] # Галогены/эфиры - Липофильность, токсичность
- ['fr_pyridine', 'fr_imidazole'] # Гетероциклы - Фармакофорные элементы
- ['fr_amide', 'fr_urea'] # Амиды - Водородные связи, основные свойства

Оценки лекарственного потенциала:

- qed: Drug-likeness (0-1)
- SPS: Синтетическая доступность

Водородные связи

Доноры/акцепторы: NumHAcceptors, NumHDonors, NHOHCount, NOCount

Специфичные группы:

fr_guanido, fr_urea - Мощные доноры/акцепторы

Циклические системы

Типы циклов: NumAliphaticCarbocycles, NumAromaticHeterocycles, NumSaturatedRings, NumAromaticRings

`Источники: https://rdkit.org/docs/py-modindex.html, https://greglandrum.github.io/rdkit-blog/posts/2022-12-23-descriptor-tutorial.html, https://teach-in.ru/file/synopsis/pdf/ai-in-chemistry-and-materials-science-M-3.pdf`

Загрузка данных, библиотек и EDA

In [5]:
```python
# График распределения по группам признаков - дискрипторов
groups = {
    "Pharmacology": 3,
    "EState": 4 + 21,
    "PhysChem": 15,
    "Topological": 18,
    "VSA": 24 + 10,
    "Electronic": 8,
    "Functional": 80,
    "DrugScore": 2,
    "HBond": 7,
    "Cycles": 6
}

plt.figure(figsize=(12, 6))
plt.bar(groups.keys(), groups.values(), color='teal')
plt.title('Распределение молекулярных дескрипторов по группам', fontsize=14)
plt.ylabel('Количество дескрипторов', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

После предварительного анализа описания признаков, выдны категориальные, числовые и неинформативные признаки, которые мы далее выделим в отдельные списки

Учитывая, что количество пропусков незначительное, а возможное искажение заполнением пропусков стандартными методами может исказить модель, удалим строки. Пропущенные значения в колонках, отражают электронную структуру и медианы тут не подойдут.

Удаление неинформативных столбцов

Видно большое количество высоко скоррелированных признаков между собой.

In [20]:
```
# Посмотрим на корреляцию числовых признаков с таргетом
for i in ['IC50, mM', 'CC50, mM', 'SI']:
    print('Пирсон:', check_corr_method(df[numeric_features], target=i))
```

```
Пирсон: IC50, mM              1.000000
CC50, mM               0.522534
BCUT2D_CHGLO           0.204597
FpDensityMorgan1       0.204462
BalabanJ               0.189869
FpDensityMorgan2       0.184937
MaxPartialCharge       0.179758
BCUT2D_MWLOW           0.159386
FpDensityMorgan3       0.154565
MinAbsPartialCharge    0.147331
dtype: float64
Пирсон: CC50, mM            1.000000
IC50, mM            0.522534
FpDensityMorgan1    0.290383
FpDensityMorgan2    0.254017
HallKierAlpha       0.213781
BCUT2D_CHGLO        0.205856
BalabanJ            0.182926
BCUT2D_LOGPLOW      0.158215
MinPartialCharge    0.156620
FractionCSP3        0.150130
dtype: float64
Пирсон: SI                  1.000000
BalabanJ            0.164715
FpDensityMorgan1    0.087894
VSA_EState4         0.087770
NHOHCount           0.079056
EState_VSA2         0.071871
SMR_VSA5            0.065880
FractionCSP3        0.063552
SlogP_VSA3          0.060063
Kappa3              0.053846
dtype: float64
```

In [21]:
```
# Посмотрим на корреляцию категориальных признаков с таргетом
for i in ['IC50, mM', 'CC50, mM', 'SI']:
    print('Пирсон:', check_corr_method(df[cat_features + bool_features + ['IC50, mM', 'CC50, mM', 'SI']], target=i, metho
```

```
Пирсон: IC50, mM                   1.000000
CC50, mM                    0.455493
NumSaturatedHeterocycles    0.222683
NumAliphaticHeterocycles    0.174191
fr_alkyl_halide             0.145538
NumAromaticHeterocycles     0.137250
fr_Ndealkylation2           0.109475
fr_nitro                    0.102639
fr_furan                    0.100815
fr_halogen                  0.099047
dtype: float64
Пирсон: CC50, mM                   1.000000
IC50, mM                    0.455493
fr_Imine                    0.138547
NumSaturatedHeterocycles    0.135363
fr_alkyl_halide             0.101475
fr_quatN                    0.090466
fr_furan                    0.086571
fr_Nhpyrrole                0.085859
fr_Ar_NH                    0.085859
fr_C_S                      0.085097
dtype: float64
Пирсон: fr_Imine                   0.176811
NumSaturatedCarbocycles     0.156249
NumAliphaticCarbocycles     0.142418
fr_Ndealkylation1           0.102927
fr_NH2                      0.095620
fr_quatN                    0.090523
fr_guanido                  0.070429
CC50, mM                    0.070308
fr_priamide                 0.069321
fr_Al_OH_noTert             0.060343
dtype: float64
```

Между таргетами и числовыми, категориальными признаками присутствует слабая коррелиция/ранговая зависимость (соответственность).

По предварительной оценке Линейная модель не будет иметь большого успеха в предсказании нужных параметров соединения. В этой связи я не буду удалять признаки с высокой корреляцией, так как на деревянные модели, нейросети это не оказывает влияния, а лишь добавляет вычислительной сложности.
Использование методов типа PCA также не целесообразно для подобных моделей, т.к. это ухудшит описательную способность результатов. Для лекарственных препаратов это намного важнее чем, результат из черной коробки.

In [22]:
```python
# Выбросы по числовым признакам
X_n = df.copy()

# Инициализируем DataFrame для отметок выбросов по каждому столбцу
outlier_flags = pd.DataFrame(index=X_n.index)

# Проверяем выбросы в каждом столбце
for column in X_n[numeric_features].columns:
    z_scores = np.abs(stats.zscore(X_n[column]))
    outlier_flags[column] = z_scores >= 3

# Объединяем результаты: True, если есть выброс хотя бы в одном столбце
X_n['is_outlier'] = outlier_flags.any(axis=1)
X_n['is_outlier'].value_counts()
```

Out[22]:
```
is_outlier
False    533
True     465
Name: count, dtype: int64
```

In [23]:
```python
# Выбросы по таргетам
X_n = df.copy()

# Инициализируем DataFrame для отметок выбросов по каждому столбцу
outlier_flags = pd.DataFrame(index=X_n.index)

# Проверяем выбросы в каждом столбце
for column in X_n[['IC50, mM', 'CC50, mM', 'SI']].columns:
    z_scores = np.abs(stats.zscore(X_n[column]))
    outlier_flags[column] = z_scores >= 3

# Объединяем результаты: True, если есть выброс хотя бы в одном столбце
X_n['is_outlier'] = outlier_flags.any(axis=1)
X_n['is_outlier'].value_counts()
```

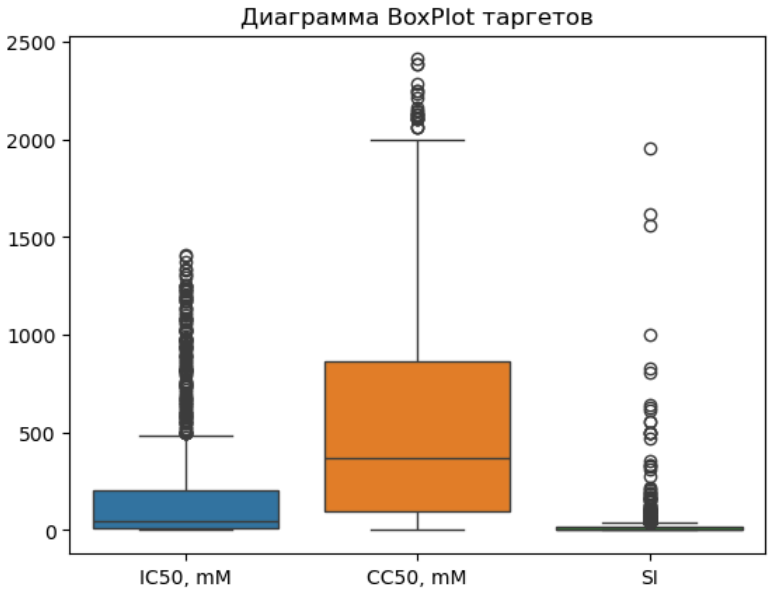
Out[23]:
```
is_outlier
False    970
True      28
Name: count, dtype: int64
```

In [24]:
```
X_n = X_n[X_n['is_outlier']==False]
```

In [25]:
```
X_n.shape
```

Out[25]: (970, 197)

In [26]:
```
# Бокс plot таргетов
sns.boxplot(X_n[['IC50, mM', 'CC50, mM', 'SI']])
plt.title("Диаграмма BoxPlot таргетов");
```



Проанализировав количество выбросов, я пришел к выводу, что удалять выбросы будем по таргетам, т.к. потеря половины выборки критична.

Инжиниринг данных

In [27]:
```
# Собрал различные вариации новых признаков , смысл котрых нашли в интернете знающие люди
def add_engineered_features(df):

    df = df.copy()

    # 1. Комбинированные физико-химические свойства
    df['LogP_MW_Ratio'] = df['MolLogP'] / (df['MolWt'] + 1e-6)
    df['PolarSurfaceFraction'] = df['TPSA'] / (df['LabuteASA'] + 1e-6)
    df['Charge_Asymmetry'] = df['MaxPartialCharge'] - df['MinPartialCharge']

    # 2. Топологические индексы
    df['Complexity_Index'] = df['BertzCT'] * df['HallKierAlpha'] / (df['NumRotatableBonds'] + 1)

    # 3. Электронные свойства
    df['BCUT_Charge_Ratio'] = df['BCUT2D_CHGHI'] / (abs(df['BCUT2D_CHGLO']) + 1e-6)
    df['EState_Range'] = df['MaxEStateIndex'] - df['MinEStateIndex']

    # 4. Функциональные группы
    polar_groups = ['fr_COO', 'fr_Ar_OH', 'fr_Al_OH', 'fr_NH2', 'fr_amide']
    lipo_groups = ['fr_halogen', 'fr_alkyl_halide', 'fr_unbrch_alkane', 'fr_aryl_methyl']

    df['Polar_Groups_Count'] = df[polar_groups].sum(axis=1)
    df['Lipophilic_Groups_Count'] = df[lipo_groups].sum(axis=1)
    df['PLB'] = df['Polar_Groups_Count'] / (df['Lipophilic_Groups_Count'] + 1e-6)

    # 5. Водородные связи
    df['HBond_Capacity'] = df['NumHDonors'] + df['NumHAcceptors']
    df['HBond_Donor_Acceptor_Ratio'] = df['NumHDonors'] / (df['NumHAcceptors'] + 1e-6)

    # 6. Циклические системы
    df['Saturation_Index'] = df['NumSaturatedRings'] / (df['RingCount'] + 1e-6)

    # 7. Производные от VSA
    hydrophobic_vsa = [f'SlogP_VSA{i}' for i in range(1, 7)]
    if set(hydrophobic_vsa).issubset(df.columns):
        df['Hydrophobic_VSA'] = df[hydrophobic_vsa].sum(axis=1)

    # 8. BCUT взаимодействия
    df['BCUT_Electronic'] = df['BCUT2D_MWHI'] * df['BCUT2D_CHGHI']

    # 9. Дополнительные найденные варианты
```

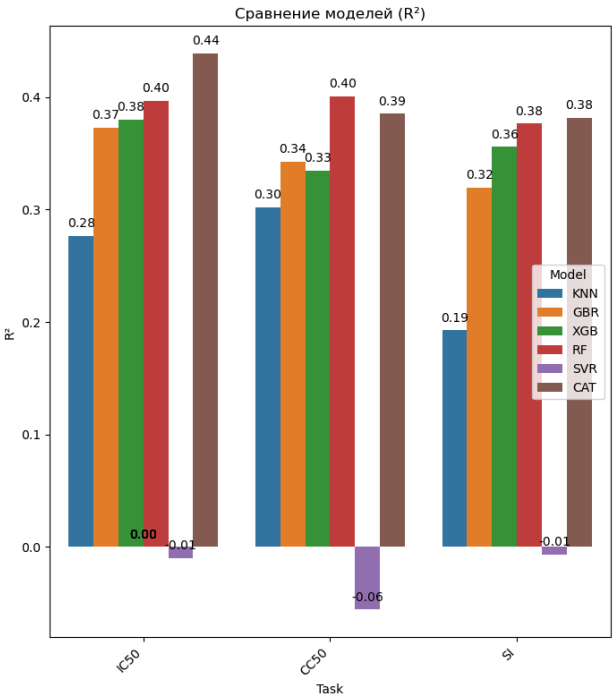
```
    df['Size_Flexibility'] = df['MolWt'] * df['NumRotatableBonds'] / 100
    df['LogD'] = df['MolLogP'] - np.log10(df['Polar_Groups_Count'] + 1)

    return df
```

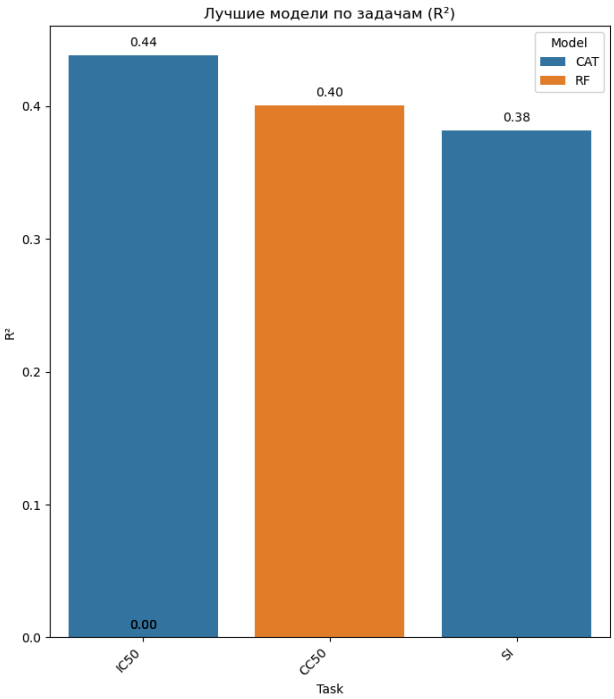
Далее я проверял, стоит ли удалять высокоскоррелированные признаки, новые признаки и как это влияет на качество наиболее предпочтительных моделей (высокой и даже средней кореляции у признаков с таргетами нет, так, что я не стал использвать линейные модели).

In [52]:
```
#  Посмотрим на результаты моделей
def print_and_plot_best_results(results_dict, metric_name, metric_label):

    tasks = []
    best_models = []
    best_scores = []
    all_data = []

    print(f"\nЛУЧШИЕ МОДЕЛИ ({metric_label}):")
    for task, models in results_dict.items():
        best_model = None
        best_score = -np.inf if metric_name != 'MSE' else np.inf
        task_data = []

        for model_name, scores in models.items():
            for subtask, metrics in scores.items():
                if metric_name in metrics:
                    score = metrics[metric_name]
                    task_data.append((model_name, score))

                    if (metric_name != 'MSE' and score > best_score) or \
                       (metric_name == 'MSE' and score < best_score):
                        best_score = score
                        best_model = model_name

        if best_model:
            tasks.append(task)
            best_models.append(best_model)
            best_scores.append(best_score)
            print(f"{task}: {best_model} ({best_score:.2f})")

            for model, score in task_data:
                all_data.append({
                    'Task': task,
                    'Model': model,
                    'Metric': metric_label,
                    'Score': score
                })

    # Визуализация результатов
    if all_data:
        plt.figure(figsize=(14, 8))

        plt.subplot(1, 2, 1)
        df_all = pd.DataFrame(all_data)
        ax = sns.barplot(x='Task', y='Score', hue='Model', data=df_all, )
        plt.title(f'Сравнение моделей ({metric_label})')
        plt.xticks(rotation=45, ha='right')
        plt.ylabel(metric_label)

        for p in ax.patches:
            ax.annotate(f"{p.get_height():.2f}",
                        (p.get_x() + p.get_width() / 2., p.get_height()),
                        ha='center', va='center',
                        xytext=(0, 10),
                        textcoords='offset points')

        # Лучшие модели
        plt.subplot(1, 2, 2)
        df_best = pd.DataFrame({
            'Task': tasks,
            'Model': best_models,
            'Score': best_scores
        })
        ax = sns.barplot(x='Task', y='Score', hue='Model', data=df_best, dodge=False)
        plt.title(f'Лучшие модели по задачам ({metric_label})')
        plt.xticks(rotation=45, ha='right')
        plt.ylabel(metric_label)

        # Добавляем значения на столбцы
        for p in ax.patches:
            ax.annotate(f"{p.get_height():.2f}",
                        (p.get_x() + p.get_width() / 2., p.get_height()),
                        ha='center', va='center',
                        xytext=(0, 10),
                        textcoords='offset points')
```

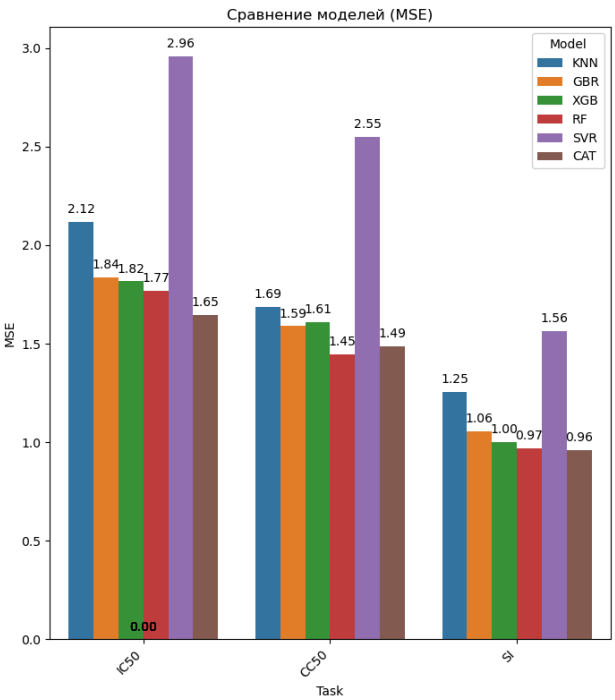
```
        plt.tight_layout()
        plt.show()


# Анализ регрессионных моделей
print_and_plot_best_results(reg_results, 'R2', 'R²')
print_and_plot_best_results(reg_results, 'MSE', 'MSE')

# Анализ классификационных моделей
print_and_plot_best_results(clf_results, 'F1', 'F1-score')
print_and_plot_best_results(clf_results, 'AUC', 'AUC')
```

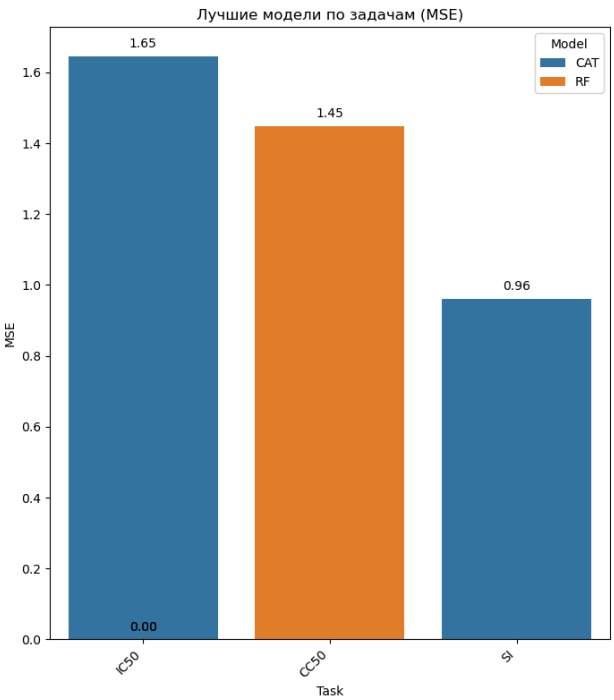
```
ЛУЧШИЕ МОДЕЛИ (R²):
IC50: CAT (0.44)
CC50: RF (0.40)
SI: CAT (0.38)
```



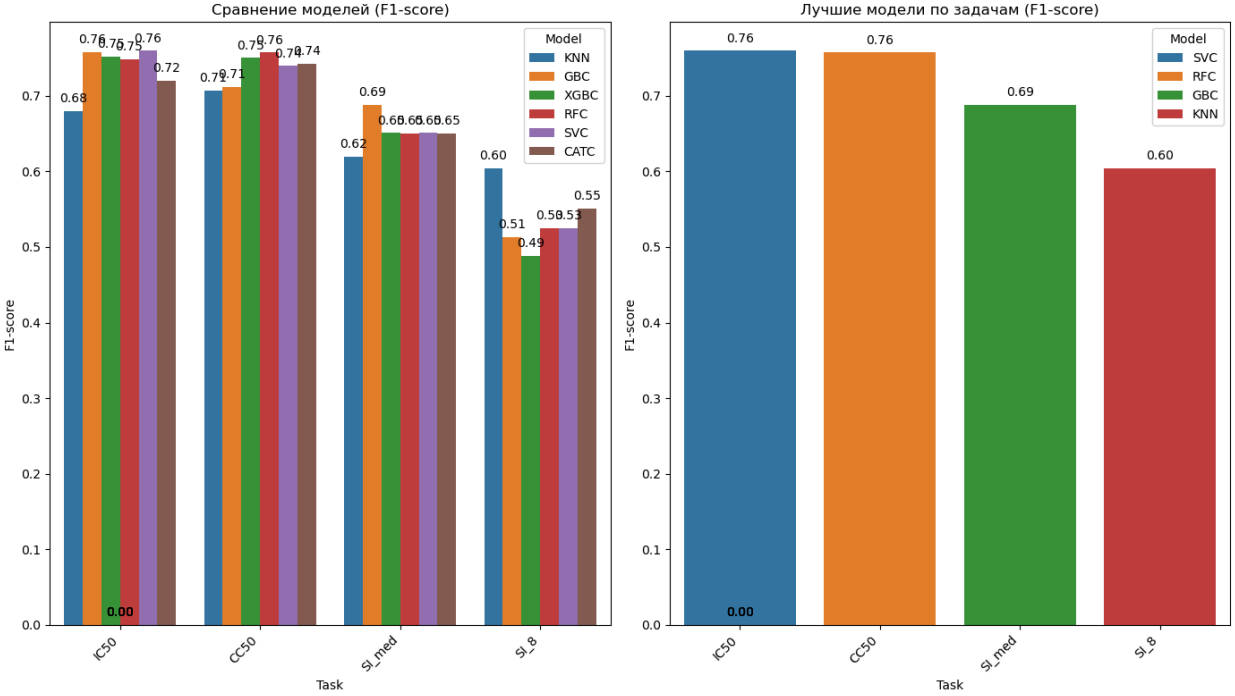



```
ЛУЧШИЕ МОДЕЛИ (MSE):
IC50: CAT (1.65)
CC50: RF (1.45)
SI: CAT (0.96)
```

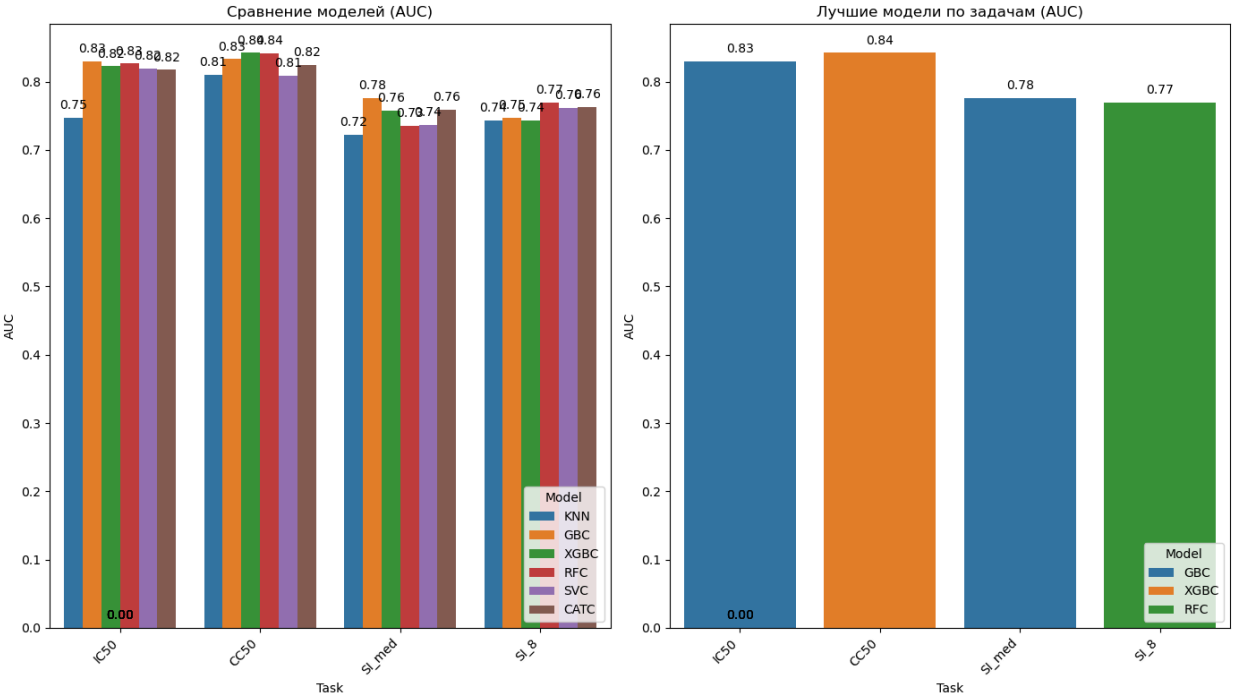




```
ЛУЧШИЕ МОДЕЛИ (F1-score):
IC50: SVC (0.76)
CC50: RFC (0.76)
SI_med: GBC (0.69)
SI_8: KNN (0.60)
```




```
ЛУЧШИЕ МОДЕЛИ (AUC):
IC50: GBC (0.83)
CC50: XGBC (0.84)
SI_med: GBC (0.78)
SI_8: RFC (0.77)
```



Итоговый вывод по результатам EDA и анализа данных:

- в предоставленном ДатаСете 1001 запись, 214 признаков из них 1 отвечает за индексы, остальные молеклярные дискрпторы, отражающие определенные параметры соединения.
- 3 записи с пропущенными значениями, заполнение медианой или средним не целесообразно, ввиду возможного вляния на ошибки первого и воторго рода. Я их удалил.

В датасете:

- категориальные признаки с булевыми значениями - 30
- обычные категориальные - 48

- числовые признаки - 118, 1 индексный
- неинформативные - 18, удалены.
- У признаков присутствует высокая корреляция между собой и слабая корреляция с таргетом. Учитывая общую слабую линейную зависимость от таргетов, использование Линейных (неполимиальных) моделей и удаление признаков не целесообразно.
- Анализ аномальных значений и выбросов показал, большое количество выбросов в данных. При удалении выбросов по всем столбцам по правилу 3 сигм теряется до половины выборки. Дальнейшие тесты показали, что при их удалении качество модели не растет. Т.к. деревянные модели хорошо справляются с подобными явлениями. Я иные незначительно повышают метрику. Так, что принято решение отфильтровать 28 выбросов по таргетам.

EDA окончен, далее приступаем к доработке лучших моделей под список задач.

Результаты обучения, в отдельных файлах:

#	Наименование проекта	Описание	Стек	Результат
1.	Задача классификации: превышает ли значение SI медианное значение выборки	Задача классификации: превышает ли значение SI медианное значение выборки	python, pandas, numpy, scipy, sklearn, matplotlib, catboost, pytorch, GB, LR	GB - roc_auc - 0.71
2.	Задачи регрессии для расчета индекса SI	Построение регрессионной модели для расчета показателя селективности химического соединения	python, pandas, numpy, scipy, sklearn, matplotlib, catboost, pytorch	CatBoost - rmse - 0.98
3.	Задачи классификации индекса SI > 8 для отбора перспективных соединений	Построение классификационной модели для отбора химического соединения с высоким показателем селективности	python, pandas, numpy, scipy, sklearn, matplotlib, catboost, pytorch, GB, LR	Нейронная сеть - roc_auc - 0.81
4.	Задачи регрессии IC50	Построение модели для оценки концентрации, необходимой для подавления вирусной репликации на 50%	python, pandas, numpy, scipy, sklearn, matplotlib, catboost, pytorch	CatBoost - rmse - 1,39
5.	Задача классификации: превышает ли значение IC50 медианное значение выборки	Построение модели для оценки концентрации, необходимой для подавления вирусной репликации на 50%, классификация	python, pandas, numpy, scipy, sklearn, matplotlib, catboost, pytorch, GB, LR	Нейронная сеть - roc_auc - 0.82
6.	Задачи регрессии CC50, mM	Построение модели для оценки концентрации, вызывающая гибель 50% здоровых клеток	python, pandas, numpy, scipy, sklearn, matplotlib, catboost, pytorch	Нейронная сеть - rmse - 1.23
7.	Задача классификации: превышает ли значение CC50 медианное значение выборки	Построение модели для оценки концентрации, вызывающая гибель 50% здоровых клеток, классификация	python, pandas, numpy, scipy, sklearn, matplotlib, catboost, pytorch, GB, LR	Нейронная сеть - roc_auc - 0.81

Итоговые результат:

Все полученные по результатам решения задач модели достигли показателей выше медианных/рандомных/линейных подходов к предсказанию будущего результата. Однако требуют дополнительной доработки в части, удаления мультиколлинеарных признаков, которыми можно пожертвовать без вреда для здоровья человека. Созданию новых агрегированных признаков, которые также могут повысить точность моделей.

К сожалению, без профильного образования и возможности консультации со специалистами, сложно принять решение по EDA и Feature engeniring.

Вместе с тем, хоть полученные модели могут быть доработаны, даже данные базовые версии могут быть полезны и сократить рутинный труд при создании химических соединений для борьбы с противовирусными препаратами, но не могут служить однозначными мерилом качества соединения, т.к. имеют достаточно высокий процент ошибок 1 и 2 рода.

Проведенная работа по исследованию данных химических соединений показала, что модели машинного обучения могут быть использованы для отбора перспективных соединений, оценки в случае генерации соединений алгоритмами.

Что качается стека технологий, то нейронные сети являются наиболее перспективным подходом в построении разделяющих поверхностей и регрессионных моделей. По моему мнению усложнение архитектуры нейронной сети поможет увеличить качество моделей. Что требует дальнейшего исследования.

Источники:

- https://rdkit.org/docs/py-modindex.html,
- https://greglandrum.github.io/rdkit-blog/posts/2022-12-23-descriptor-tutorial.html,
- https://teach-in.ru/file/synopsis/pdf/ai-in-chemistry-and-materials-science-M-3.pdf

In [ ]: