

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

Марат Нұрсұлтан Олжасұлы

Өмір сүру салалары үшін машиналық оқыту әдістерін қолдану

**ДИПЛОМДЫҚ ЖҰМЫС**

5B070300 - «Ақпараттық жүйелер» мамандығы

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

КАӨЖС кафедрасы меңгерушісі  
техн. ғыл. канд., ассистент-  
профессор

\_\_\_\_\_ Н.А. Сейлова  
«\_\_» \_\_\_\_\_ 2020 ж.

**ДИПЛОМДЫҚ ЖҰМЫС**

Тақырыбы: «Өмір сүру салалары үшін машиналық оқыту әдістерін қолдану»

5B070300 - «Ақпараттық жүйелер» мамандығы

Орындаған

Марат Н.

Ғылыми жетекші  
техн.ғыл.магистр., лектор  
\_\_\_\_\_ А.Н.Дуйсенбаева  
«\_\_» \_\_\_\_\_ 2020 ж.

Алматы 2020

# ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

5B070300 - Ақпараттық жүйелер

## БЕКІТЕМІН

КАӨЖС кафедрасы меңгерушісі  
техн. ғыл. канд., ассистент-  
профессор

\_\_\_\_\_ Н.А. Сейлова  
«\_\_» \_\_\_\_\_ 2020 ж.

## Дипломдық жұмысты орындауға ТАПСЫРМА

Білім алушы Марат Нұрсұлтан Олжасұлы

Жобаның тақырыбы: «Өмір сүру салалары үшін машиналық оқыту әдістерін қолдану»

Университет Ректорының «27» қаңтар 2020 жылғы №762-б бұйрығымен бекітілген.

Орындалған жұмыстың өткізу мерзімі «29» мамыр 2020 ж.

Дипломдық жұмыстың бастапқы мәліметтері: берілген тақырып бойынша әдебиеттерге шолу кезінде жиналған мәліметтер, теориялық материалдар жинау.

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

- а) пәндік саланы талдау;
- ә) машиналық оқыту моделін құру;
- б) программалық қамтаманы құру.

Графикалық материалдардың тізімі (міндетті түрде қажет сызбалар көрсетілген): жұмыстың \_\_\_ слайдтан тұратын презентациясы көрсетіледі.

Негізгі әдебиет 11 кітаптан тұрады.

Дипломдық жұмысты даярлау  
**КЕСТЕСІ**

| Бөлім атаулары,<br>қарастырылатын<br>мәселелер тізімі | Ғылыми жетекшіге,<br>кеңесшілерге өткізу<br>мерзімі | Ескерту |
|---|---|---------|
| Пәндік саланы талдау                                  | 10.01.2020 - 07.02.2020                             |         |
| Машиналық оқыту<br>моделін құру                       | 08.02.2020 - 09.03.2020                             |         |
| Программалық<br>қамтаманы құру                        | 10.03.2020 - 27.04.2020                             |         |

Дипломдық жұмыс бөлімдерінің кеңесшілері мен  
норма бақылаушының аяқталған жұмысқа қойған  
**қолтаңбалары**

| Бөлімдердің<br>атауы | Кеңесшілер<br>(аты-жөні, тегі,<br>ғылыми дәрежесі,<br>атағы) | Қол қойылған<br>мерзімі | Қолы |
|----------------------|--|-------------------------|------|
| БҚ жасау             | А.Н.Дуйсенбаева<br>техн.ғыл.магистр.,<br>лектор              |                         |      |
| Норма<br>бақылаушы   |  |                         |      |

Ғылыми жетекшісі \_\_\_\_\_ А.Н.Дуйсенбаева

Тапсырманы орындауға қабылдаған білім алушы \_\_\_\_\_ Н. Марат

Күні «27» қаңтар 2020 ж.

**Протокол анализа Отчета подобия Научным руководителем**

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Марат Нурсултан

**Название:** Использование методов машинного обучения для областей жизнедеятельности

**Координатор:** Асемгуль Дуйсенбаева

**Коэффициент подобия 1:3,7**

**Коэффициент подобия 2:3**

**Замена букв:3**

**Интервалы:0**

**Микропробелы:0**

**Белые знаки: 0**

**После анализа Отчета подобия констатирую следующее:**

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....

.....  
*Дата*

.....  
*Подпись Научного руководителя*

**Протокол анализа Отчета подобия**

**заведующего кафедрой / начальника структурного подразделения**

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Марат Нурсултан

**Название:** Использование методов машинного обучения для областей жизнедеятельности

**Координатор:** Асемгуль Дуйсенбаева

**Коэффициент подобия 1:3,7**

**Коэффициент подобия 2:3**

**Замена букв:3**

**Интервалы:0**

**Микропробелы:0**

**Белые знаки:0**

**После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:**

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

**Обоснование:**

.....  
.....  
.....  
.....  
.....  
.....

Дата

*Подпись заведующего кафедрой /*

*начальника структурного подразделения*

**Окончательное решение в отношении допуска к защите, включая обоснование:**

.....  
.....  
.....  
.....  
.....  
.....

Дата

.....  
*Подпись заведующего кафедрой /  
начальника структурного подразделения*

DOCUMENT.REPORT.SIMILARITY.REPORT-DATE: 2020-05-22 12:44:54

DOCUMENT.REPORT.SIMILARITY.TITLE:

Использование методов машинного обучения для областей жизнедеятельности

DOCUMENT.REPORT.SIMILARITY.AUTHOR:

Марат Нурсултан

DOCUMENT.REPORT.SIMILARITY.COORDINATOR:

Асемгүль Дуйсенбаева

DOCUMENT.REPORT.SIMILARITY.ORGANISATIONAL-UNIT:

ИКиИТ

DOCUMENT.REPORT.SIMILARITY.UPLOAD-DATE:

2020-05-22 12:41:57

DOCUMENT.REPORT.SIMILARITY.NUMBER-OF-CHECKS:

1

DOCUMENT.REPORT.SIMILARITY.DOCUMENT-SKIP-ADDRESSES:

## document.report.similarity.record-of-similarities

document.report.similarity.record.of.similarities.info



## document.report.similarity.warnings-and-alerts

document.report.similarity.warnings-and-alerts.info

|   |   |   |
|---|---|---|
| document.report.similarity.alert<br>document.report.similarity.alert.info                       | 3 | document.report.similarity.show-in-the-text |
| document.report.similarity.spreads<br>document.report.similarity.spreads.info                   | 0 | document.report.similarity.show-in-the-text |
| document.report.similarity.micro-spaces<br>document.report.similarity.micro-spaces.info         | 0 | document.report.similarity.show-in-the-text |
| document.report.similarity.white-characters<br>document.report.similarity.white-characters.info | 0 | document.report.similarity.show-in-the-text |

## document.report.similarity.active-lists-of-similarities

document.report.similarity.active.lists.info

### document.report.similarity.top-ten-fragments (3,66 %)

document.report.similarity.top.ten.fragments.info

| DOCUMENT.REPORT.SIMILARITY.LP | DOCUMENT.REPORT.SIMILARITY.TITLE-OR-SOURCE  | DOCUMENT.REPORT.SIMILARITY.AUTHOR | DOCUMENT.REPORT.SIMILARITY.WORDS.COPIED |
|-------------------------------|---|-----------------------------------|---|
| 1                             | <a href="https://stud.kz/referat/show/92623">https://stud.kz/referat/show/92623</a>         | 86                                | 2,28 %                                  |
| 2                             | <a href="https://stud.kz/prezentatsiya/id/31888">https://stud.kz/prezentatsiya/id/31888</a> | 28                                | 0,74 %                                  |
| 3                             | <a href="https://stud.kz/referat/show/92623">https://stud.kz/referat/show/92623</a>         | 18                                | 0,48 %                                  |
| 4                             | <a href="https://stud.kz/referat/show/92623">https://stud.kz/referat/show/92623</a>         | 6                                 | 0,16 %                                  |

### document.report.similarity.from-ref-books-database (0,00 %)

document.report.similarity.from.ref.books.database.info

| DOCUMENT.REPORT.SIMILARITY.LP               | DOCUMENT.REPORT.SIMILARITY.TITLE | DOCUMENT.REPORT.SIMILARITY.AUTHOR | DOCUMENT.REPORT.SIMILARITY.WORDS.COPIED |
|---|----------------------------------|-----------------------------------|---|
| DOCUMENT.REPORT.SIMILARITY.NO.SOURCES.FOUND |                                  |                                   |   |

### document.report.similarity.from-the-home-base (0,00 %)

document.report.similarity.from.the.home.base.info

| DOCUMENT.REPORT.SIMILARITY.LP               | DOCUMENT.REPORT.SIMILARITY.TITLE | DOCUMENT.REPORT.SIMILARITY.AUTHOR | DOCUMENT.REPORT.SIMILARITY.DATE | DOCUMENT.REPORT.SIMILARITY.WORDS-FRAGMENTS |
|---|----------------------------------|-----------------------------------|---------------------------------|--|
| DOCUMENT.REPORT.SIMILARITY.NO.SOURCES.FOUND |                                  |                                   |                                 |  |

### document.report.similarity.from-the-base-exchange-program (0,00 %)

document.report.similarity.from.the.base.exchange.program.info

| DOCUMENT.REPORT.SIMILARITY.LP               | DOCUMENT.REPORT.SIMILARITY.TITLE | DOCUMENT.REPORT.SIMILARITY.AUTHOR | DOCUMENT.REPORT.SIMILARITY.DATE | DOCUMENT.REPORT.SIMILARITY.WORDS-FRAGMENTS |
|---|----------------------------------|-----------------------------------|---------------------------------|--|
| DOCUMENT.REPORT.SIMILARITY.NO.SOURCES.FOUND |                                  |                                   |                                 |  |

### document.report.similarity.from-the-internet (3,66 %)

document.report.similarity.from.the.internet.info

| DOCUMENT.REPORT.SIMILARITY.LP | DOCUMENT.REPORT.SIMILARITY.SOURCE-URL   | DOCUMENT.REPORT.SIMILARITY.IDENTICAL-WORDS-FRAGMENTS |
|-------------------------------|---|--|
| 1                             | <a href="https://stud.kz/referat/show/92623">https://stud.kz/referat/show/92623</a>         | 110 (3) 2,92 %                                       |
| 2                             | <a href="https://stud.kz/prezentatsiya/id/31888">https://stud.kz/prezentatsiya/id/31888</a> | 28 (1) 0,74 %  |



## АНДАТПА

Дипломдық жобаның тақырыбы: «Өмір сүру салалары үшін машиналық оқыту әдістерін қолдану».

Дипломдық жұмыстың көлемі 44 бет, оның ішінде 45 сурет пайдаланылған. Түйінді сөздер: Машиналық оқыту, Data Mining, web қосымша.

Дипломдық жұмыстың негізгі мақсаты - Алматы қаласындағы сатылатын пәтер бағаларын зерттеп, зерттеу нәтижесінде машиналық оқыту құралдарын пайдаланып, пәтер параметрлері арқылы пәтер бағасын болжайтын регрессиялық модель құру.

## **АННОТАЦИЯ**

Тема дипломного проекта: «Использование методов машинного обучения для областей жизнедеятельности».

Объем дипломной работы 44 страницы, в том числе 45 рисунок. Ключевые слова: машинное обучение, Data Mining, web приложение .

Основной целью дипломной работы является изучение цен на квартиру в г. Алматы, а также создание регрессионной модели с использованием машинного обучения и прогнозом стоимости квартиры по параметрам квартиры.

## **ANNOTATION**

Theme: «Using machine learning methods in life areas».

The volume of the thesis is 44 pages, including 45 drawings. Keywords: machine learning, Data Mining, web application .

The main goal of the thesis is to study the prices of apartments in Almaty, as well as to create a regression model using machine learning and forecasting the cost of apartments by apartment parameters.

## МАЗМҰНЫ

|   |    |
|---|----|
| КІРІСПЕ   | 9  |
| 1 Пәндік аймаққа шолу                             | 10 |
| 1.1 Әзірленетін жүйенің өзектілігі                | 10 |
| 1.2 Ұқсас тапсырмаларда машиналық оқытуды қолдану | 11 |
| 2 Әдістер мен алгоритмдерге шолу                  | 14 |
| 2.1 Қолданылған әдістердің сипаттамасы            | 14 |
| 2.2 Қолданылған алгоритмдердің сипаттамасы        | 17 |
| 2.3 Қолданылған технологиялар және құралдар       | 19 |
| 3 Практикалық бөлім                               | 21 |
| 3.1 Деректерді жинау                              | 21 |
| 3.2 Деректерді өңдеу және анализ                  | 22 |
| 3.3 Random forest арқылы оқыту моделін құру       | 30 |
| 3.4 Болжам жасау процесі және нәтиже шығару       | 33 |
| 3.5 Веб қосымшаны әзірлеу                         | 35 |
| ҚОРЫТЫНДЫ   | 38 |
| ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ                   | 39 |
| А қосымшасы                                       | 40 |

## КІРІСПЕ

Қазіргі уақытта біз «деректер дәуірінде» өмір сүрудеміз, көптеген деректер күн сайын жиналады және сақталады. Осы өсіп келе жатқан деректер алдында машиналық оқыту әдістері сөзсіз қажет болды.

Машиналық оқыту - аналитикалық моделді құруды автоматтандыратын деректерді талдау әдісі. Бұл жүйелер деректер негізінде оқып, заңдылықтарды анықтап, адамның ең аз араласуымен шешім қабылдай алатын идеяға негізделген жасанды интеллект саласы. Классикалық алгоритмдерге тапсырманы орындау үшін нақты және толық ережелер берілген. Машиналық оқыту алгоритмдеріне модельді, сондай-ақ деректерді анықтайтын жалпы ұсынымдар берілген. Бұл деректер міндетті орындау үшін модельге қажетті жеткіліксіз ақпаратты қамтуы тиіс. Осылайша, Машиналық оқыту алгоритмі модель деректерді ескере отырып түзетілген кезде өз міндетін орындай алады. Біз оны «модельді деректерге келтіреміз» немесе «модель деректерге оқытылуы тиіс»деп айтамыз.

Жалпы, машиналық оқыту біз қолмен кодталған болуы тым күрделі ақпарат немесе ақпарат толық емес кезде күрделі тапсырмалар үшін өте пайдалы. Мұндай жағдайларда біз модельге қолда бар ақпаратты беру арқылы оған қажет ақпаратты оқытуға мүмкіндік береміз. Содан кейін алгоритм мәліметтерден тікелей жетіспейтін білім алу үшін статистикалық әдістерді қолданады.

Машиналық оқыту - бұл ауруларды диагностикалаудан бастап климаттың жаһандық өзгеруі үшін шешімдерді іздестіруге дейінгі мәселелерді шешу және кең ауқымды мәселелер бойынша негізделген қорытынды жасау қабілетін арттыру үшін пайдалануға болатын құрал.

Дипломдық жұмыстың негізгі мақсаты - Алматы қаласындағы сатылатын пәтер бағаларын зерттеп, зерттеу нәтижесінде машиналық оқыту құралдарын пайдаланып пәтер параметрлері арқылы пәтер бағасын болжайтын регрессиялық модель құру.

## **1 Пәндік аймаққа шолу**

### **1.1 Әзірленетін жүйенің өзектілігі**

Біздің еліміздің нарықтық экономикаға көшуі ғылым мен тәжірибенің бірқатар жаңа салаларын, атап айтқанда, жылжымайтын мүлік құнын бағалауды терең дамытуды талап етті. Бағалау-бұл жылжымайтын мүлік объектісінің құны туралы кәсіпқойдың әдіснамалық дәлелденген дүниетанымы және оны анықтау процесі.

Жылжымайтын мүлік объектілерінің нарықтық құнын анықтау күрделі және бірегей процесс болып табылады, себебі мүлдем ұқсас екі объектіні табу мүмкін емес. Тіпті ғимараттар бір типтік жоба бойынша салынғандығымен, бірақ әртүрлі жер учаскелерінде орналасқан жағдайда олардың құны айтарлықтай өзгеше болуы мүмкін.

Жылжымайтын мүлікпен жасалатын барлық операциялар мен мәмілелер жылжымайтын мүлік объектісінің құнын білуді талап етеді. Нарықтық жағдайларда жылжымайтын мүліктің құны жалпы қоғам экономикасы мен өміріндегі факторларға, үрдістерге және өзгерістерге байланысты. Жылжымайтын мүлікті бағалау қажеттілігі мынадай жағдайларда туындайды:

- сатып алу - сату немесе жалға беру операциялары;
- кәсіпорындарды акционерлеу және мүліктік үлестерді қайта бөлу;
- жаңа пайшыларды және акциялардың қосымша эмиссиясын тарту;
- жылжымайтын мүлік объектілеріне: ғимараттар мен жер учаскелеріне салық салу мақсаттары үшін кадастрлық бағалау;
- жылжымайтын мүлік объектілерін сақтандыру;
- жылжымайтын мүлік объектілерін кепілге салу;
- кәсіпорындар мен ұйымдардың жарғылық капиталына салым ретінде жылжымайтын мүлік объектілерін енгізу;
- инвестициялық жобаларды әзірлеу және инвесторларды тарту;
- жылжымайтын мүлік объектілерін жою;
- мұрагерлік құқықтарын, сот үкімін орындау, мүліктік дауларды шешу;
- жылжымайтын мүлік объектілеріне мүліктік құқықтарды іске асыруға байланысты басқа да операциялар.

Қазақстан Республикасының Азаматтық кодексіне сәйкес: «Жылжымайтын мүлікке (жылжымайтын заттар) жер учаскелері, ғимараттар, құрылыстар, көпжылдық екпелер және жермен тығыз байланысты өзге мүлік, яғни орнынан олардың мақсатына сай емес шығынсыз ауыстыру мүмкін болмайтын объектілер жатады».

### **1.2 Ұқсас тапсырмаларда машиналық оқытуды қолдану**

Денсаулық сақтау. Қазіргі уақытта Машиналық оқыту пациенттерді жедел диагностикалау және жедел медициналық көмек көрсету үшін қолданылады.

Әкімшіліктен, жазбаларды жүргізу толыққанды диагностика мен емдеуге дейін, ML дағдарысты талдау және оны дұрыс емдеу мен процедура үшін көптеген басқа сценарийлермен салыстыру мүмкіндігі бар. Бұл салыстыру уақытты үнемдейді және шешуші медициналық тәсіл үшін стратегиялық жолды салады. Сонымен қатар, Машиналық оқыту хирургиялық роботтарға дәрігерлерге медициналық процедураларда көмек көрсете алады, бұл ретте ең аз басып кіруді және жоғары дәлдікті қамтамасыз етеді. Бұл жетістік хирургиялық процедуралардың табыстылық көрсеткіштерін жақсартып, жұмысты экономикалық пайдалармен орындау мерзімін тездетуі мүмкін. Денсаулық сақтау контекстінде ML тиімді диагностика, зерттеу және емдеу үшін шешуші фактор бола алады, бұл сектордың тұтас өзгеруін көрсетеді.

Көлік саласы. Бүгінгі көлік саласы машиналық оқытудың күшті ықпалында. Технология сенсорлық басқару мен автоматтандыруды өрістетуді арқылы ақылсыз жүргізуден шығатын қауіпті жоюда маңызды рөл атқарды. Бұл көлік құралдарына қоршаған параметрлерді түсінуге және жолаушылардың қауіпсіздігін қамтамасыз ету үшін қажет болған кезде сақтық шараларын қолдануға мүмкіндік берді. Қытай өз қаласы үшін жүргізушісіз автобустарды сынауда, Ал Google және Rolls Royce 2020 жылға қарай әлемдегі алғашқы өздігінен жүретін кемелі бірлесіп жасап, іске қосты. Бұл кеме теңізді және оның қоршаған объектілерін түсіну үшін Google Cloud Machine Learning Engine пайдаланады. Бұл кемелі қашықтықтан басқаруға және тасымалдауға тартылған ресурстарды қысқартуға мүмкіндік береді.

BFSI-банктік және қаржылық қызмет. Бірнеше банктер мен қаржы мекемелері кредиттік тәуекелдерді талдау және болжау және қосымшаның сапасын бағалау үшін машиналық оқыту негізінде кешенді алгоритмдерді пайдаланады, бұл оларға негізделген шешімдер қабылдауға мүмкіндік береді. Бұл сондай-ақ сарапшылар көмегімен алаяқтықты анықтауға көмектеседі. Дауысты тану, чат-боттар және болжамдық талдау қаржы мекемелері мен әлеуетті клиенттер арасындағы алшақтықты жеңуге көмектеседі. Бұл уақытты үнемдейді, сондай-ақ ұйымдарда клиенттерге қызмет көрсету сапасын жақсартады.

Білім беру. Бұл мұғалімдер боттармен ауыстырылуы мүмкін емес, бірақ оларға Машиналық оқыту олардың оқыту және жалпы оқыту әдістерін қамтамасыз ете алатын әртүрлілікке көмектесе алады. Машиналық оқыту мүмкіндіктері баланың академиялық түсінігін бағалау, оның білімін қалай қабылдайтынын талдау және осы нақты оқушының пайдалы қасиеттеріне назар аударуға көмектесетін жеке академиялық жоспар құру үшін қолданылады. Алгоритмдер тест нәтижелерін талдайды және мұғалімдердің уақытын босатып, балаларға арналған оқу модульдерін жақсартуға көмектесетін бірегей бағалау жүйесін құрады. ML мүмкіндігі шектеулі және оқыту кемшіліктері бар студенттерге өте пайдалы болуы мүмкін.

Заң. Машиналық оқыту заң қызметкерлерінің күнделікті қызметін автоматтандыру арқылы оларды артық міндеттерден босата алады және оларға қаралатын істер үшін жауапты шешімдер жасауға көңіл бөлуге көмектесе алады.

Машина оқыту осы автоматтандыруды қамтамасыз етеді. Ол құжаттарды толықтырады, оларды өңдейді, сондай-ақ оларды дәлелдемелер мен зерттеулерге талдайды, олар адвокаттарға кітаптар мен заң құжаттарын парақтауға уақыт жұмсамай, тиісті ақпаратты алуға көмектесе алады. Олар сондай-ақ тыңдаудан хабардар болу және алдағы күндердің есебін жүргізу сияқты жедел міндеттерді жүйелендіруге көмектесе алады.



## 2 Әдістермен алгоритмдерге шолу

### 2.1 Қолданылған әдістердің сипаттамасы

*Машиналық оқыту түсінігіне шолу.* Машиналық оқыту жасанды интелект элементтерінің бірі болып табылады, соның арқасында компьютер өзін - өзі оқыту және белгілі бір тапсырманы орындауды жақсарту қабілетіне бағытталған. Шын мәнінде, Машиналық оқыту үлкен деректерді талдауға - ақпаратты автоматты түрде алуға және оны болжау үшін пайдалануға, болжамның дұрыстығын анықтауға және, егер жоқ болса, болашақта дәл болжам жасау үшін осының негізінде оқытуға негізделеді.

Google, Amazon, Netflix және басқа да монолитті онлайн платформалар іздеу тарихын талдау семантикалық нәтижелерді ұсыну машиналық оқыту алгоритмдеріне негізделген. Бұл ірі компаниялар күн сайын миллиардтаған деректермен жұмыс жасайды. Бұл күн сайын жиналатын үлкен деректерді машиналық оқыту көмегімен талдау мүмкін емес еді.

Машиналық оқыту (Machine Learning) - жасанды интелектің кең ауқымды бөлімшесі, алгоритмдерді құру әдістерін зерттейді. Оқытудың екі түрі бар. Прецеденттер бойынша оқыту немесе индуктивті оқыту жеке эмпирикалық деректер бойынша жалпы заңдылықтарды анықтауға негізделген. Дедуктивтік оқыту сарапшылардың білімін формалдауды және оларды білім базасы түрінде компьютерге көшіруді көздейді. Сондықтан Машиналық оқыту және прецеденттер бойынша оқыту терминдерін синонимдер деп есептеуге болады.

Машиналық оқыту математикалық статистиканың, оңтайландыру әдістері мен классикалық математикалық пәндердің түйіскен жерінде орналасқан, бірақ есептеу тиімділігі мен қайта оқыту проблемаларымен байланысты өзіндік ерекшелігі бар. Индуктивті оқытудың көптеген әдістері классикалық статистикалық тәсілдерге балама ретінде әзірленген. Көптеген әдістер ақпаратты алумен және деректерді интеллектуалды талдаумен тығыз байланысты (Data Mining).

Машиналық оқытудың теориялық бөлімдері жеке бағытқа, есептеу оқыту теориясына (Computational Learning Theory, COLT) біріктірілген.

Машиналық оқыту - математикалық ғана емес, практикалық, инженерлік пән. Таза теория, әдетте, практикада қолданылатын әдістер мен алгоритмдерге бірден әкелмейді. Оларды жақсы жұмыс істеуге мәжбүр ету үшін теорияда жасалған болжамдардың нақты есептер шарттарына сәйкес еместігін өтейтін қосымша эвристиканы ойлап табуға тура келеді. Машиналық оқытудағы бірде-бір зерттеу әдістің практикалық жұмысқа қабілеттілігін растайтын модельдік немесе нақты деректерде экспериментсіз өтпейді.

*Прецеденттер бойынша оқыту есептерінің жалпы қойылымы.* Прецедент туралы деректер оның сипаттамасы деп аталады. Прецеденттердің барлық сипаттамаларының жиынтығы оқыту үлгісі деп аталады. Осы жеке деректер бойынша осы нақты іріктемеге ғана тән жалпы тәуелділікті, заңдылықтарды,

өзара байланысты анықтаумен қоса, жалпы барлық прецеденттерге, соның ішінде әлі байқалмаған прецеденттерге де анықтау талап етіледі.

Прецеденттерді сипаттаудың ең көп таралған жолы белгі болып табылады. Барлық прецеденттерде өлшенетін көрсеткіштердің N жиынтығы тіркеледі. Прецеденттер уақытша қатармен немесе сигналдармен, суреттермен, мәтіндермен, ұқсастықтың қосарланған қатынастарымен немесе өзара іс-қимыл қарқындылығымен сипатталған және т. б. неғұрлым күрделі жағдайлар болуы мүмкін.

Прецеденттер бойынша оқыту міндеттерін шешу үшін бірінші кезекте қалпына келтірілетін тәуелділік моделі тіркеледі. Содан кейін сапа функционалы енгізіледі, оның мәні модель байқалатын деректерді қаншалықты жақсы сипаттайтынын көрсетеді. Оқыту алгоритмі берілген оқыту үлгісінде сапа функционалы оңтайлы мәнге ие модель параметрлерінің жиынтығын іздейді. Деректерді таңдау бойынша модельді реттеу процесі (fitting) көп жағдайда сандық оңтайландыру әдістерін қолдануға әкеледі.

*Мұғаліммен оқыту түсінігі.* Мұғаліммен оқыту (Supervised learning) - келесі міндетті шешуге арналған машиналық оқыту бөлімдерінің бірі. Көптеген нысандар (жағдайлар) және көптеген ықтимал жауаптар (Пікірлер, реакциялар) бар. Жауаптар мен объектілер арасында кейбір тәуелділік бар, бірақ ол белгісіз. Тек қана прецеденттердің соңғы жиынтығы белгілі - «объект, жауап» жұбы, оқыту үлгісі деп аталады. Бұл деректер негізінде тәуелділікті қалпына келтіру қажет, яғни кез келген объект үшін жеткілікті дәл жауап беруге қабілетті алгоритм салу. Жауаптардың дәлдігін өлшеу үшін сапа функционалы белгілі бір түрде енгізіледі.

*Data mining түсінігі.* Data mining - кең түсінік. Бүгінде оның ондаған анықтамалары бар. Жалпы мағынада деректерді талдау - есептеумен байланысты зерттеулер көптеген параметрлерге ие көп өлшемді деректер жүйесі. Деректерді талдау барысында зерттеуші осы деректермен сипатталатын құбылыстың сипаты туралы белгілі бір түсініктерді қалыптастыру мақсатында іс-әрекеттер жиынтығын жүргізеді. Әдетте, деректерді талдау үшін әртүрлі математикалық әдістер қолданылады. Деректерді талдауды тек ақпаратты өңдеу ретінде қарастыруға болмайды. Деректерді талдау - бұл ең алдымен, гипотезаны тексеру және зерттеуші есептерін шешу құралы.

«Модель» сөзі (лат. *modellum*) «Өлшем», «тәсіл», « қандай - онда затты». Модельдерді құру - тәуелділікті анықтауға, болжауға, топтарға бөлуге және басқа да көптеген міндеттерді шешуге мүмкіндік беретін қоршаған ортаны зерттеудің әмбебап тәсілі. Модельдеудің негізгі мақсаты модельдеуші жүйенің жұмысын жақсы көрсету керек. Модель - түпнұсқаны жақсы зерделеу немесе оның қандай да бір қасиеттерін жаңғырту үшін бір жүйені басқа жүйемен алмастыру үшін объектінің, жүйенің объектісі немесе сипаттамасы. Модельдеу - білімді алудың, сипаттау мен пайдаланудың әмбебап әдісі.

Модельдеу түрі бойынша модель бөлінеді:

- эмпирикалық - эмпирикалық фактілер, тәуелділік негізінде алынған;
- теориялық - математикалық сипаттамалар, заңдар негізінде алынған;

- полуэпирикалық - эмпирикалық тәуелділік және математикалық сипаттамалар негізінде алынған.

Деректерге анализ жасау - бұл шешім қабылдау үшін ұйымдастырылған деректерден ақпаратты алу әдістері мен құралдарының жиынтығы. Анализ жасау - оны алғаннан кейін және жинағаннан кейін ақпаратты өңдеу ғана емес, бұл гипотезаны тексеру құралы. Деректерді кез келген талдаудың мақсаты - зерттелетін жағдайды толығымен түсіну (үрдістерді, соның ішінде жоспардан жағымсыз ауытқуларды анықтау, болжамдау және ұсынымдарды алу). Осы мақсатқа жету үшін келесі деректерді талдау міндеттері қойылады:

- ақпарат жинау,
- ақпаратты құрылымдау,
- заңдылықтарды анықтау, талдау,
- ұсыныстарды болжау және алу.

Барлық деректер маңызды ақпараттан тұрады, бірақ әр түрлі сұрақтар үшін. Барлық массивтен нақты мән-жайлар үшін пайдалы деректерді бөлу үшін оларды өңдеу қажет.

Деректерді өңдеу процесінде анализге дайындау жүргізіледі, оның нәтижесінде олар шешілетін есептің ерекшелігімен анықталатын талаптарға сәйкес келтіріледі.

Алдын ала өңдеу Data Mining маңызды кезеңі болып табылады және егер ол орындалмаса, одан кейінгі анализде көптеген жағдайларда аналитикалық алгоритмдер жұмысына кедергі болатынды немесе олардың жұмысының нәтижелері дұрыс емес болуы мүмкін. Басқаша айтқанда, GIGO - garbage in, garbage out (кіре берістегі қоқыс, шыға берістегі қоқыс) принципі іске асырылады.

Деректерді өңдеу екі бағытты қамтиды: тазалау және оңтайландыру. Тазалау деректер сапасын төмендететін және аналитикалық алгоритмдердің жұмысына кедергі келтіретін әртүрлі факторларды болдырмау мақсатында жүргізіледі. Ол дубликаттарды, қарама-қайшылықтарды және жалған мәндерді өңдеуді, рұқсатнамаларды қалпына келтіру мен толтыруды, шуды басуды және ауытқушылық мәндерді редакциялауды қамтиды. Сонымен қатар, тазарту барысында құрылымның бұзылуы, деректердің толықтығы мен тұтастығы қалпына келтіріледі, қате форматтар өзгертіледі.

Деректерді оңтайландыру алдын ала өңдеу элементі ретінде өлшемдікті азайту, білмейтін белгілерді анықтау және алып тастау кіреді. Тазалаудан оңтайландырудың негізгі айырмашылығы тазалау процесінде жойылатын факторлар тапсырманы шешудің дәлдігін айтарлықтай төмендетеді немесе аналитикалық алгоритмдердің жұмысын мүмкін емес етеді. Оңтайландыру кезінде шешілетін мәселелер деректерді нақты міндетке бейімдейді және оларды талдаудың тиімділігін арттырады.

Деректерді алдын ала өңдеу және тазалау - бұл модельді оқыту үшін деректер жинағын пайдаланбас бұрын орындалуы қажет маңызды міндеттер. Өңделмеген деректер жиі бұрмаланған және сенімсіз, және оларда мәндерді жіберіп алмауы мүмкін. Модельдеу кезінде мұндай деректерді пайдалану дұрыс

емес нәтижелерге әкелуі мүмкін. Бұл есептер деректер тобын өңдеу және талдау процесінің бір бөлігі болып табылады және әдетте қажетті алдын ала өңдеуді анықтау және жоспарлау үшін пайдаланылатын деректер жинағын бастапқы зерттеуді білдіреді.

Нақты деректер әр түрлі көздер мен процестерден кейін өңдеу үшін жиналады. Олар деректер жиынтығының сапасына теріс әсер ететін қателер мен зақымдарды қамтуы мүмкін. Бұл деректер сапасына типтік проблемалар болуы мүмкін:

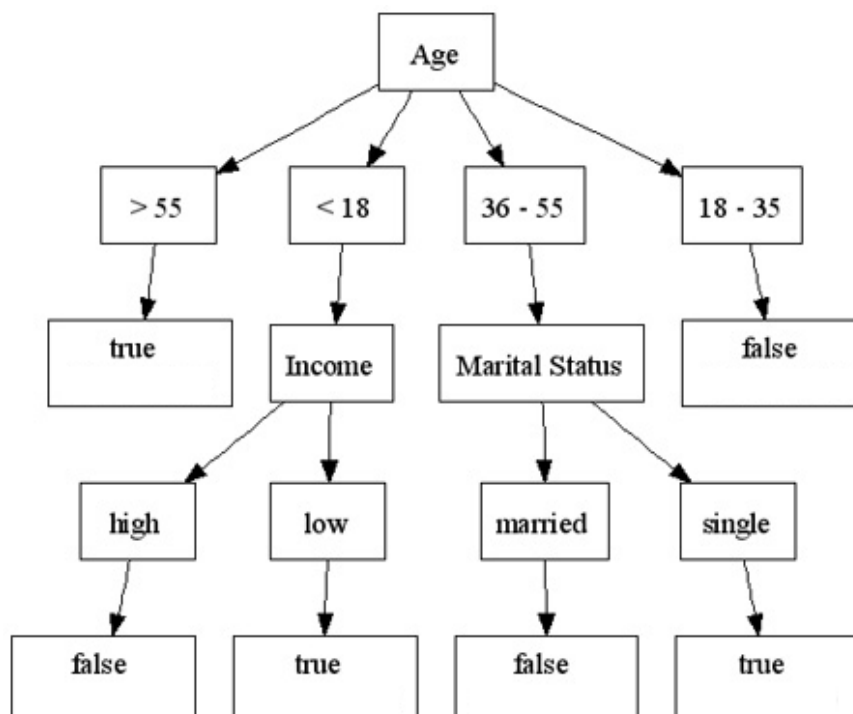
Толық емес: деректерде атрибуттар жоқ немесе оларда мәндер жоқ.

Шу: деректер қате жазбалар немесе шығарындылар бар.

Келіспеушілік: деректер өзара қақтығысатын жазбалардан немесе алшақтықтардан тұрады.

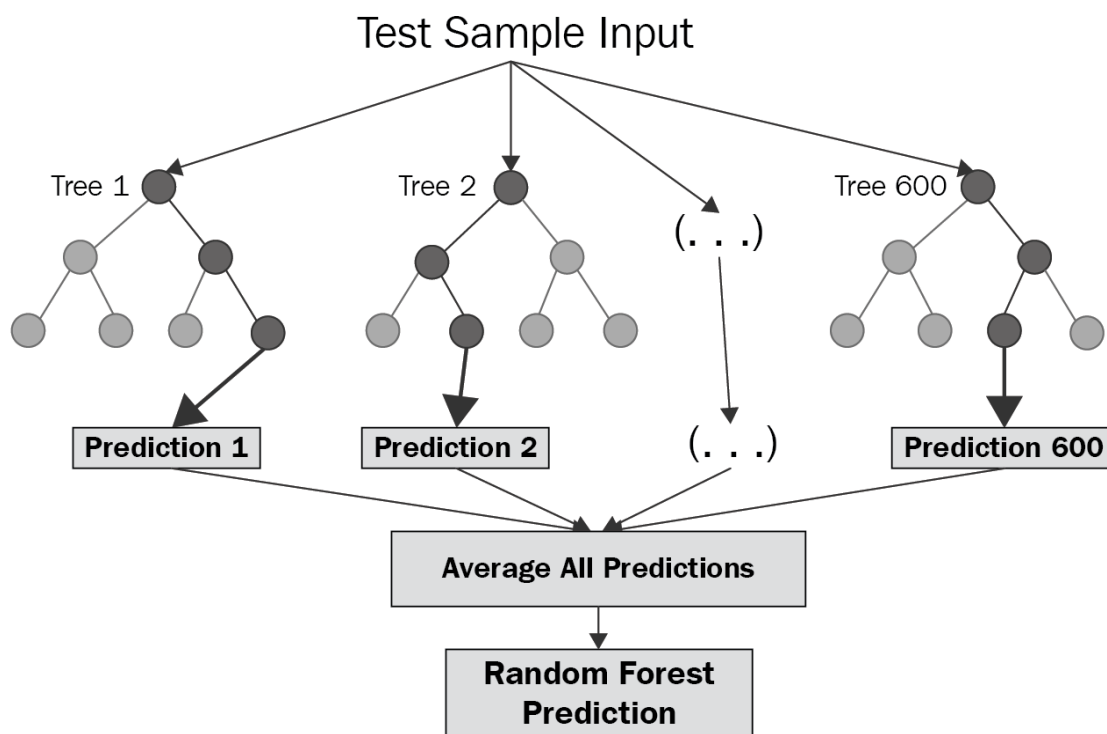
## 2.2 Қолданылған алгоритмдердің сипаттамасы

*Decision tree және random forest.* Decision tree - классификация және регрессия есептерін шешетін классификацияның логикалық алгоритмі. Логикалық шарттарды ағаш құрылымына біріктіру болып табылады. Кестелік деректермен жақсы жұмыс істейтін машиналық оқытудың ең танымал алгоритмдерінің бірі. Оның негізгі идеясы - іріктемені барынша ұсақ бөліктерге бөліп алу, соңында мысалдарды дұрыс жиындар бойынша барынша дәл біріктіру.



2.2.1 Сурет - Decision tree

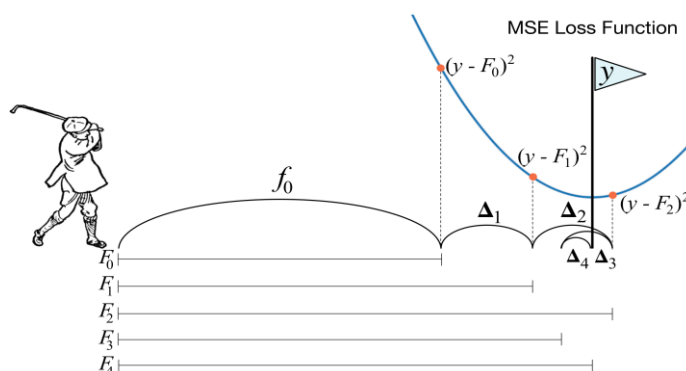
Random forest - бұл көптеген шешуші ағаштарның (decision tree) жиынтығынан тұрады. Регрессия есебінде олардың жауаптары орташаланады, классификация есебінде көпшілік бойынша дауыс беру арқылы шешім қабылданады.



### 2.2.2 Сурет - Random forest

XGBoost - градиенттік бустинг алгоритмінің ең танымал және тиімді жүзеге асырылуының бірі. Модельдердің сапасына қарсы күрестегі келесі қадам алдыңғы шеңберде жасалған қателіктерге ерекше назар аудара отырып, «қатемен жұмыс істеуді» жасап және модельдің алдыңғы нұсқасын жете меңгере отырып, ағаштарды салумен көп циклдық кіруді жасауға мүмкіндік беретін градиентті бустинг болды. XGBoost негізінде шешім ағаштарының градиенттік бустинг алгоритмі жатыр.

Градиентті бустинг - бұл жіктеу және регрессия есептері үшін машиналық оқыту техникасы, ол ансамбль түрінде болжау моделін жасайды. Ансамбльді оқыту, мысалы, бэггингке қарағанда жүйелі түрде жүргізіледі. Әрбір итерацияда оқыту үлгісінде оқытылған ансамбльдің болжамдарының ауытқулары есептеледі. Ансамбльге қосылатын келесі модель осы ауытқуларды болжайды. Осылайша, оқытылған ансамбльдің болжамдарына жаңа ағаштың болжамдарын қосу арқылы біз моделінің орташа ауытқуын азайтуға болады. Жаңа ағаштар қате азайғанша немесе «ерте тоқтату» ережелерінің бірі орындалғанға дейін ансамбльге қосылады.



2.2.3 Сурет - Бустинг иллюстрациясы

### 2.3 Қолданылған технологиялар және құралдар

Деректерді жинау процесінде Java программалау тілі және JSOUP кітапханасы қолданылды.

Data mining және болжам жасау Python программалау тілінде Jupyter Notebook программасы қолданылады. Jupyter Notebook - (бұрын IPython Jupyter Notebook ретінде белгілі) - Python, Julia, R, Haskell және Ruby бағдарламалау тілдері үшін интерактивті орта. Бұл есептеу, графиктер, кестелер, сондай - ақ статистикалық модельдеу және машиналық оқыту үшін ыңғайлы құрал. Jupyter ендірілген интерактивті қорытындыға ие, соның арқасында өңдеу ортасында бейнені, бейнені, динамикалық графиканы және т.б. жасауға болады. Келесі кітапханалар анализ және болжам жасау процесстерінде қолданылды:

- NumPy - бұл үлкен көп өлшемді массивтер мен матрицалардың қолдауын қосатын Python тілінің кітапханасы, осы массивтермен операциялар үшін жоғары деңгейлі (және өте жылдам) математикалық функциялардың үлкен кітапханасы.

- pandas - деректерді өңдеу және талдау үшін Python тіліндегі бағдарламалық кітапхана. Деректер pandas жұмысы төмен деңгейдегі құрал болып табылатын NumPy кітапханасының үстінен құрылады. Сандық кестелермен және уақытша қатармен айла-шарғы жасау үшін арнайы деректер құрылымы мен операцияларды ұсынады.

- Scikit - learn - классикалық машиналық оқыту міндеттерін шешу үшін ең көп таралған таңдау. Ол мұғаліммен және мұғалімсіз оқыту алгоритмдерінің кең таңдауын ұсынады. Мұғаліммен оқыту мақсатты белгінің мәні белгіленген датасеттің болуын көздейді. Мұғалімсіз оқыту датасетте белгі қоюды көздемейді - еркін деректерден пайдалы ақпаратты алуды үйрену қажет етеді. Кітапхананың негізгі артықшылықтарының бірі ол бірнеше кең таралған математикалық кітапханалар негізінде жұмыс істейді және оларды бір-бірімен оңай біріктіреді.

- Web қосымша Python программалау тілінің Flask фреймворк арқылы жасалды. Flask-веб-қосымшаларды жасау үшін арналған микроплатформа. Flask веб-қосымшаны жасау үшін қажетті барлық құралдарды, кітапханаларды және технологияларды ұсынады.

## 3 Практикалық бөлім

### 3.1 Деректерді жинау

Машиналық оқытудың ең бастапқы әрі маңызды кезеңі деректерді жинау болып табылады. Деректерді жинау - бұл қойылған зерттеу сұрақтарына жауап беруге, гипотезаларды тексеруге және нәтижелерді бағалауға мүмкіндік беретін белгіленген жүйелі түрде қызығушылық танытқан айнаымалылар бойынша ақпаратты жинау және өлшеу процесі. Зерттеу деректерін жинау компоненті физикалық және әлеуметтік ғылымдар, гуманитарлық ғылымдар, бизнес және т.б. қоса алғанда, оқытудың барлық салалары үшін ортақ болып табылады. Зерттеу саласына немесе деректерді (сандық, сапалық) анықтаудың қалауына қарамастан, деректерді нақты жинау зерттеудің тұтастығын сақтау үшін маңызды. Деректерді жинаудың тиісті құралдарын таңдау (бар, өзгертілген немесе жаңадан әзірленген), сондай-ақ оларды дұрыс пайдалану жөніндегі нақты шектелген нұсқаулықтар да қателердің туындау ықтималдығын төмендетеді.

Қате жиналған деректердің салдары қамтиды:

- зерттеу сұрақтарына дәл жауап бере алмау;
- зерттеуді қайталау және растау қабілетсіздігі;
- ресурстарды жоғалтуға әкелетін бұрмаланған нәтижелер.

Деректерді жинау процесінде бірыңғай өлшем бірліктерінде, дұрыс жазуда базаға жиналуға, мүмкін болатын қателерді және кодтау қателерін ескеру өте маңызды. Деректерді жинаудың ең маңызды мақсаты бай ақпарат пен сенімді деректердің анализін жасау үшін жиналуын қамтамасыз ету болып табылады. Болжау моделінің сапасы деректердің сапасына тікелей байланысты. Мен деректерді жинауда Java программалау тілінде жазылған парсер көмегімен деректерді JSON форматында жинадым.

#### Продажа квартир в Алматы

Найдено 59 117 объявлений

Сначала: [новые объявления](#) [дешевые](#) [дорогие](#)

[Показать объявление](#)

в этот раздел



3-комнатная квартира, 67 м², 8/9 этаж

35 000 000 ₸

Наурызбайский р-н, мкр Каргалы

В залого, жил. комплекс Хан-Тенгри, 2017 г.п., состояние: евроремонт, потолки 3м., санузел совмещенный, телефон: есть

Хозяин

Алматы · 21 мая · 6682



3-комнатная квартира, 121.5 м²

от 83 835 000 ₸

Бостандыкский р-н, Сыргабекова 55

жил. комплекс Венский дом, 4 этажа, 2018 г.п., жил. площадь 83.2 кв.м., кухня 11.2 кв.м., потолки 3.15м., санузел 2 су и более,

[Новостройка](#) ЖК «Венский дом» от застройщика «Globus invest group»

#### Горячие предложения в Алматы

[1-комнатная квартира, 33 м², 4/4 этаж за 14 млн ₸](#)  
мкр Алматы, Алмагуль-Розыбакиева ултепова

[3-комнатная квартира, 91 м² за 28 млн ₸](#)  
мкр Кулагер

[2-комнатная квартира, 85 м², 9/15 этаж за 28 млн ₸](#)  
Теле би-Напротив озера Сайран

[1-комнатная квартира, 45 м², 8/17 этаж за 19 млн ₸](#)  
мкр Мамыр-1

[3-комнатная квартира, 80.8 м², 3/14 этаж за 28 млн ₸](#)  
мкр Калкман-2, ул. Абишева

### 3.1 Сурет - Деректер ресурстары

Деректер <https://krisha.kz/> сайтынан алынған Алматы қаласында сатылатын пәтерлердің бағасын, адресін, қабырға материалын, салынған жылын, пәтердің

қабатын, үйдің жалпы қабат санын, ауданын, пәтердің жай-күйін көрсететін деректерден тұрады. Деректер жалпы 10 бағаннан, 3000 қатардан тұрады.

```
{
  "price": "30 000 000 ",
  "city": "Алматы",
  "district": " Бостандыкский р-н",
  "address": " Навои 314А – Биржана",
  "wallmaterial": "кирпичный",
  "year": " 1993 ",
  "floorNumber": "5 ",
  "floorsTotal": " 5",
  "totalArea": "65 ",
  "state": "хорошее"
},
```

### 3.2 Сурет - Деректер JSON форматында

#### 3.2 Деректерді өңдеу және анализ

Деректерге анализ жасау - бұл шешім қабылдау үшін ұйымдастырылған деректерден ақпаратты алу әдістері мен құралдарының жиынтығы. Анализ жасау - оны алғаннан кейін және жинағаннан кейін ақпаратты өңдеу ғана емес, бұл гипотезаны тексеру құралы. Деректерді кез келген талдаудың мақсаты - зерттелетін жағдайды толығымен түсіну (үрдістерді, соның ішінде жоспардан жағымсыз ауытқуларды анықтау, болжамдау және ұсынымдарды алу). Осы мақсатқа жету үшін келесі деректерді талдау міндеттері қойылады:

- ақпарат жинау;
- ақпаратты құрылымдау;
- заңдылықтарды анықтау, талдау;
- ұсыныстарды болжау және алу.

Барлық деректер маңызды ақпараттан тұрады, бірақ әр түрлі сұрақтар үшін. Барлық массивтен нақты мән - жайлар үшін пайдалы деректерді бөлу үшін оларды өңдеу қажет.

Деректерді өңдеу процесінде анализге дайындау жүргізілді, оның нәтижесінде олар шешілетін есептің ерекшелігімен анықталатын талаптарға сәйкес келтірілді.

Алдын ала өңдеу Data Mining маңызды кезеңі болып табылады, және егер ол орындалмаса, одан кейінгі анализде көптеген жағдайларда аналитикалық алгоритмдер жұмысына кедергі болатынды немесе олардың жұмысының нәтижелері дұрыс емес болуы мүмкін . Басқаша айтқанда, GIGO — garbage in, garbage out (кіре берістегі қоқыс, шыға берістегі қоқыс) принципі іске асырылады.



Деректерді өңдеу екі бағытты қамтиды: тазалау және оңтайландыру. Тазалау деректер сапасын төмендететін және аналитикалық алгоритмдердің жұмысына кедергі келтіретін әртүрлі факторларды болдырмау мақсатында жүргізіледі. Ол дубликаттарды, қарама - қайшылықтарды және жалған мәндерді өңдеуді, рұқсатнамаларды қалпына келтіру мен толтыруды, шуды басуды және ауытқушылық мәндерді редакциялауды қамтиды. Сонымен қатар, тазарту барысында құрылымның бұзылуы, деректердің толықтығы мен тұтастығы қалпына келтіріледі, қате форматтар өзгертіледі.

Деректерді алдын ала өңдеу және тазалау - бұл модельді оқыту үшін деректер жинағын пайдаланбас бұрын орындалуы қажет маңызды міндеттер. Өңделмеген деректер жиі бұрмаланған және сенімсіз, және оларда мәндерді жіберіп алмауы мүмкін. Модельдеу кезінде мұндай деректерді пайдалану дұрыс емес нәтижелерге әкелуі мүмкін.

Нақты деректер әр түрлі көздер мен процестерден кейін өңдеу үшін жиналады. Олар деректер жиынтығының сапасына теріс әсер ететін қателер мен зақымдарды қамтуы мүмкін. Бұл деректер сапасына типтік проблемалар болуы мүмкін:

Толық емес: деректерде атрибуттар жоқ немесе оларда мәндер жоқ.

Шу: деректер қате жазбалар немесе шығарындылар бар.

Келіспеушілік: деректер өзара қақтығысатын жазбалардан немесе алшақтықтардан тұрады.

Біздің жағдайда деректер жинау процесінен кейінгі деректердің жалпы көрінісі келесідей болды.

| 1  | price     | city   | district        | address            | wallmaterial | year | totalArea | state              | floorNumber | floorsTotal |
|----|-----------|--------|-----------------|--------------------|--------------|------|-----------|--------------------|-------------|-------------|
| 2  | 63000000  | Алматы | Медеуский р-н   | Чайкиной 3         | кирпичный    | 1983 | 87.2      | хорошее            | 3           | 5           |
| 3  | 54300000  | Алматы | Медеуский р-н   | Достык 103         | кирпичный    | 1964 |           | 58 евроремонт      | 2           | 4           |
| 4  | 155000000 | Алматы | Медеуский р-н   | Митина 4           | монолитный   | 2018 |           | 145 евроремонт     | 4           | 7           |
| 5  | 27500000  | Алматы | Медеуский р-н   | проспект Назар     | кирпичный    | 1975 |           | 60 евроремонт      | 1           | 5           |
| 6  | 16000000  | Алматы | Алатауский р-н  | мкр Нуркент (А)    | монолитный   | 2018 |           | 70 требует ремонта | 13          | 13          |
| 7  | 15200000  | Алматы | Бостандықский   | Сатпаева 44        | кирпичный    | 1970 |           | 40 хорошее         | 2           | 3           |
| 8  | 5400000   | Алматы | Алмалинский р-  | Сейфуллина 47      | кирпичный    | 1970 |           | 18 евроремонт      | 1           | 4           |
| 9  | 22500000  | Алматы | Алмалинский р-  | Брусилковского 159 |              | 2012 |           | 52 евроремонт      | 6           | 18          |
| 10 | 22000000  | Алматы | Бостандықский   | Розыбакиева        |              | 1961 | 57.7      | хорошее            | 1           | 4           |
| 11 | 32000000  | Алматы | Бостандықский   | мкр Коктем-2 2     | кирпичный    | 1980 | 68.7      | хорошее            | 1           | 5           |
| 12 | 19900000  | Алматы | Турксибский р-н | мкр Жұлдыз-2 4     | монолитный   | 2009 |           | 62 евроремонт      | 1           | 6           |
| 13 | 50000000  | Алматы | Бостандықский   | Абая 150/230       | монолитный   | 2009 |           | 98 евроремонт      | 10          | 17          |
| 14 | 32000000  | Алматы | Бостандықский   | Мкр Коктем-2       | кирпичный    | 1980 | 68.7      | хорошее            | 1           | 5           |
| 15 | 17000000  | Алматы | Алатауский р-н  | Нуркент — Бауі     | монолитный   | 2016 |           | 44 евроремонт      | 7           | 13          |
| 16 | 14498000  | Алматы | Жетысуский р-н  | Серикова 27        | кирпичный    | 1991 |           | 38 хорошее         | 4           | 5           |

### 3.3 Сурет - Деректердің жалпы көрінісі

Ендігі кезекте осы жиналған деректерді алдын ала өңдеу процесін жасау тапсырмасы қойылды.

Ең алдымен біз геокодер көмегімен пәтерлердің мекенжайын координаталар жүйесінде яғни бойлыққа және ендікке алмастырамыз. Бұл бізге алдағы уақытта геофикалық параметрлермен жұмыс жасауға қажет. Деректер сақталған файлды келесідей оқимыз.

```
df = pd.read_csv("alldssecond - all.csv")
```

### 3.4 Сурет - CSV файлды оқу

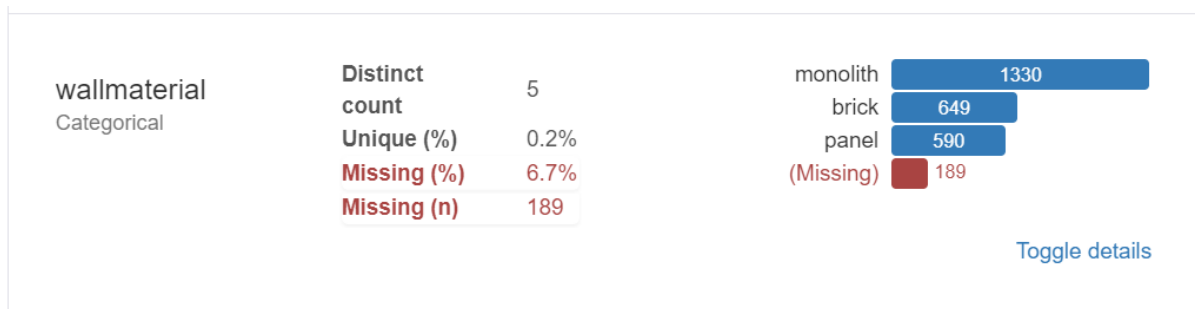
Деректер туралы ақпаратты қарайтын болсақ бізде бар параметрлердің санымен және типтері көрсетілген.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 2808 entries, 0 to 2807  
Data columns (total 9 columns):  
wallmaterial    2619 non-null object  
floorNumber     2766 non-null float64  
floorsTotal     2766 non-null float64  
totalArea       2807 non-null float64  
state           2795 non-null object  
latitude        2808 non-null float64  
longitude       2808 non-null float64  
price           2808 non-null int64  
year            2808 non-null int64  
dtypes: float64(5), int64(2), object(2)  
memory usage: 197.5+ KB
```

### 3.5 Сурет - Data Frame туралы ақпарат

wallmaterial - қабырға материалы



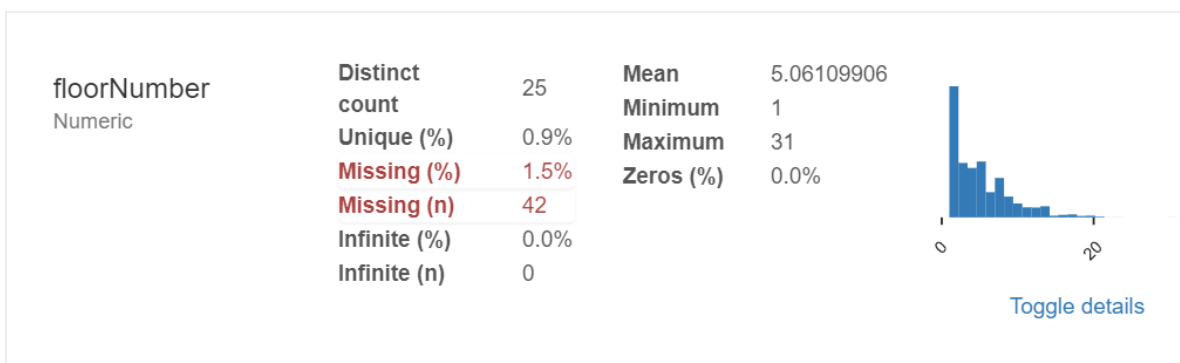
### 3.6 Сурет - wallmaterial туралы ақпарат

Жоқ деректер 6.7%-ды құрайды. Жоқ деректерді келесідей толтырамыз.

```
df.wallmaterial = df.wallmaterial.fillna('monolith')
```

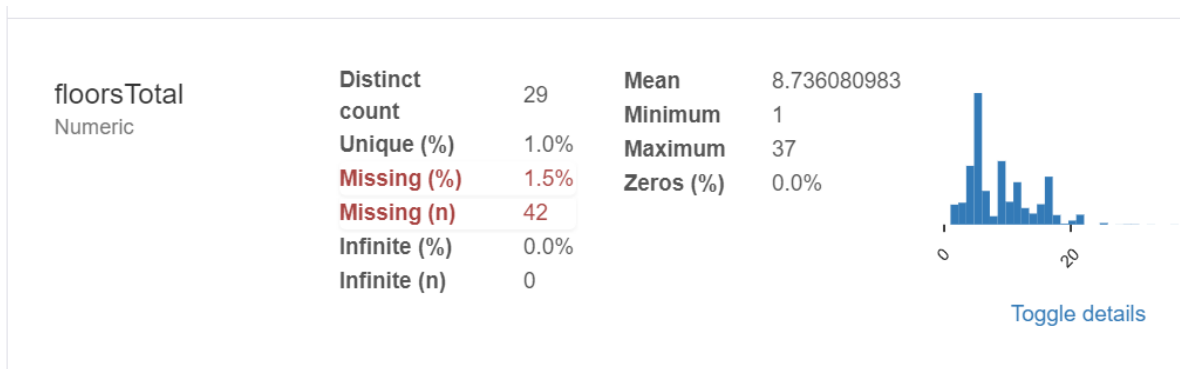
### 3.7 Сурет - wallmaterial бағанында жоқ деректерді толтыру

### floorNumber - пәтердің қабаты



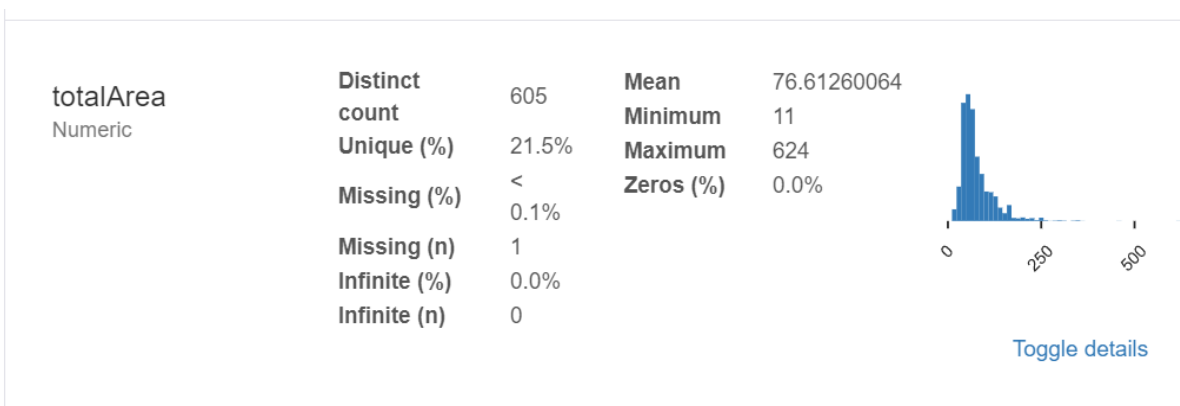
### 3.8 Сурет - floorNumber туралы ақпарат

floorsTotal - үйдің жалпы қабат саны.



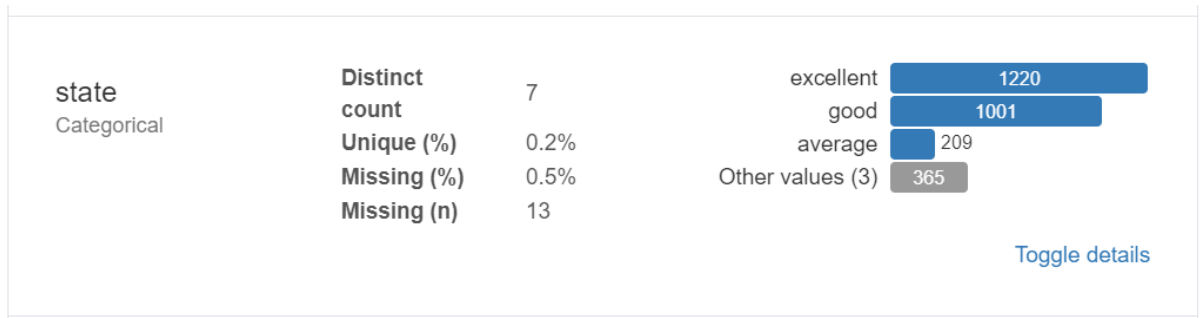
### 3.9 Сурет - floorsTotal туралы ақпарат

totalArea - пәтер ауданы



### 3.10 Сурет - totalArea туралы ақпарат

state - пәтердің жай-күйі



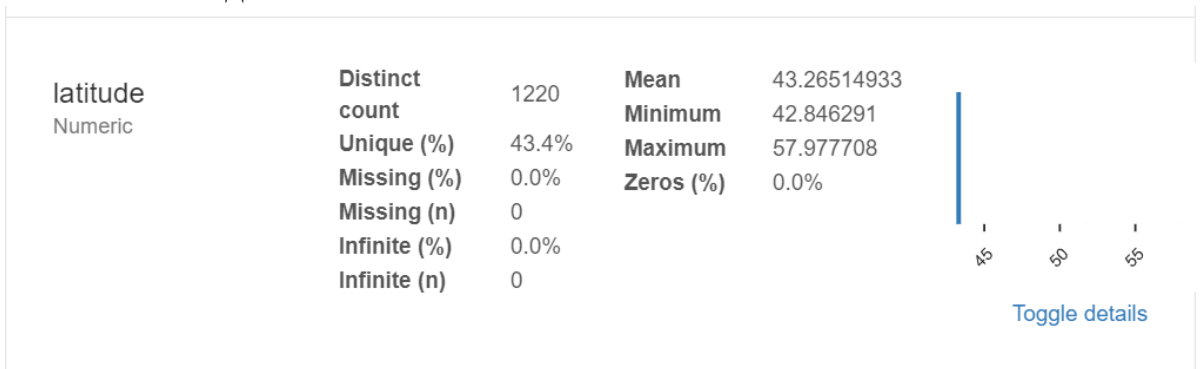
### 3.11 Сурет - state туралы ақпарат

Жоқ деректер 0.5% құрайды. Оны кездесу жиілігі орташа мәнмен толтырамыз.

```
df.state = df.state.fillna('good')
```

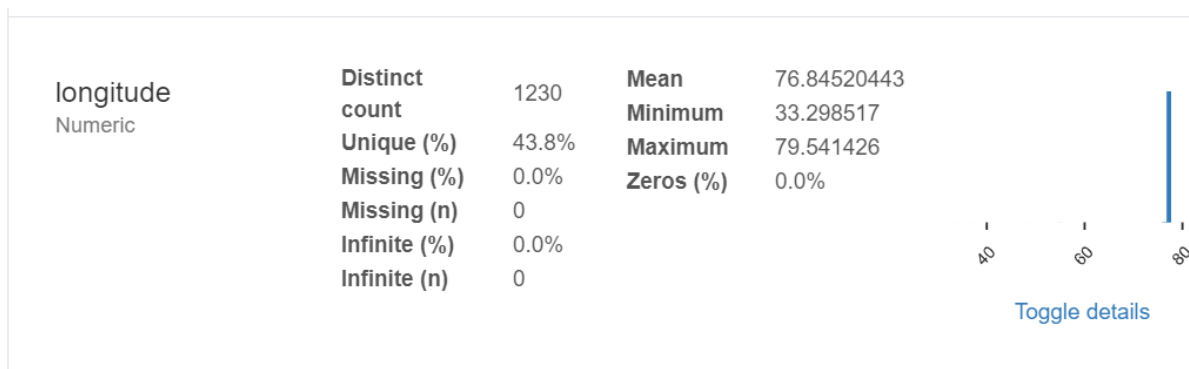
### 3.12 Сурет - state бағанында жоқ деректерді толтыру

latitude - ендік



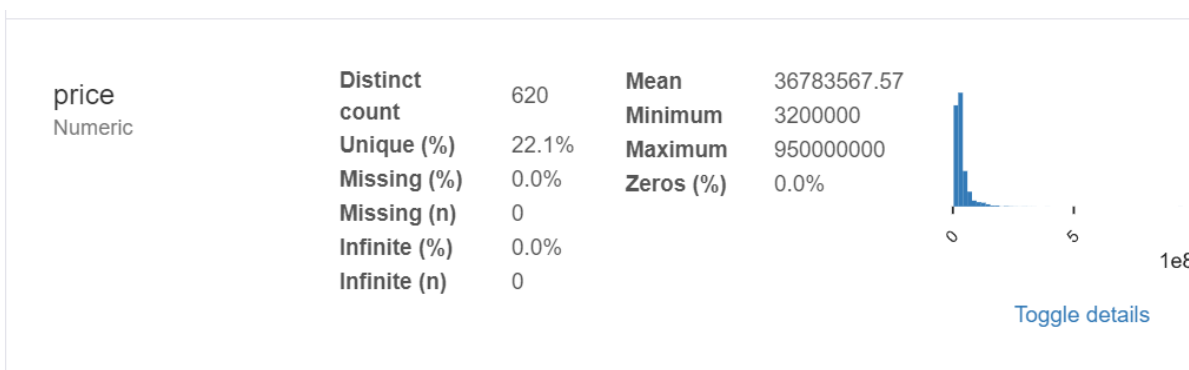
### 3.13 Сурет - latitude туралы ақпарат

longitude - бойлық



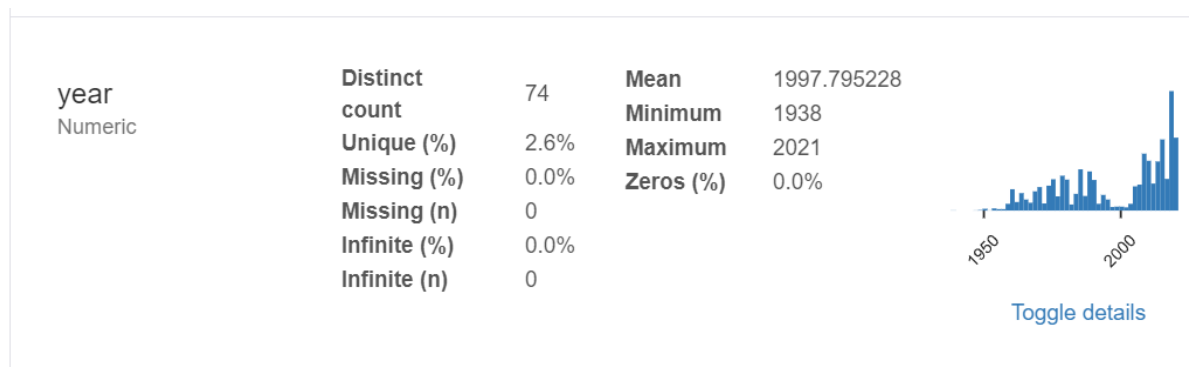
3.14 Сурет - longitude туралы ақпарат

price - пәтердің бағасы



3.15 Сурет - price туралы ақпарат

year - пәтердің салынған жылы



3.16 Сурет - year туралы ақпарат

Параметрлер генерациясы - Машиналық оқыту үшін деректерді дайындаудың ең шығармашылық кезеңі. Бұл кезең Feature Engineering деп аталады. Ол деректерді шулы ақпараттан тазалағаннан кейінгі процесс болып табылады. Feature Engineering яғни бізде бар параметрлердің негізінде жаңа

параметрлер дайындаймыз. Мен бізде бар параметрлер көмегімен 3 жаңа параметр қостым.

```
df['priceMetr'] = df['price']/df['totalArea']
```

```
city_center_coordinates = [43.238293, 76.945465]  
df['distance'] = list(map(lambda x, y: geodesic(city_center_coordinates, [x, y]).meters, df['latitude'], df['longitude']))  
df['azimuth'] = list(map(lambda x, y: get_azimuth(x, y), df['latitude'], df['longitude']))
```

### 3.17 Сурет - Feature Engineering

- priceMetr - пәтердің шаршы метрге шаққандағы ауданы;
- distance - пәтермен қала орталығына дейінгі қашықтық;
- azimuth - солтүстік бағытқа қатысты бұрыш.

Толық емес деректерді толтырғаннан кейінгі нәтижелерден көретін болсақ, бізде екі категориялық параметр бар.

|   | wallmaterial | floorNumber | floorsTotal | totalArea | state         | latitude  | longitude | price       | priceMetr    | distance     | azimuth |
|---|--------------|-------------|-------------|-----------|---------------|-----------|-----------|-------------|--------------|--------------|---------|
| 0 | brick        | 3.0         | 5.0         | 87.20     | good          | 43.215919 | 76.966198 | 63000000.0  | 7.224771e+05 | 3002.611574  | 145.97  |
| 1 | brick        | 2.0         | 4.0         | 58.00     | excellent     | 43.218166 | 76.963467 | 54300000.0  | 9.362069e+05 | 2671.836457  | 146.90  |
| 2 | monolith     | 4.0         | 7.0         | 145.00    | excellent     | 43.219007 | 76.965039 | 155000000.0 | 1.068966e+06 | 2668.221347  | 143.51  |
| 3 | brick        | 1.0         | 5.0         | 60.00     | excellent     | 43.265063 | 76.944800 | 27500000.0  | 4.583333e+05 | 2974.574105  | 358.96  |
| 4 | monolith     | 13.0        | 13.0        | 70.00     | requirerepair | 43.255675 | 76.817213 | 16000000.0  | 2.285714e+05 | 10593.307394 | 280.58  |
| 5 | brick        | 2.0         | 3.0         | 40.00     | good          | 43.235134 | 76.901726 | 15200000.0  | 3.800000e+05 | 3570.090883  | 264.35  |
| 6 | brick        | 1.0         | 4.0         | 18.00     | excellent     | 43.258183 | 76.801995 | 5400000.0   | 3.000000e+05 | 11859.164979 | 280.83  |
| 7 | monolith     | 6.0         | 18.0        | 52.00     | excellent     | 43.245399 | 76.876933 | 22500000.0  | 4.326923e+05 | 5621.904571  | 278.12  |

### 3.18 Сурет - категориялық мәндері бар параметрлер

Бұл параметрлермен жұмыс жасау үшін оларды сан мәндерге ауыстырдым.

```
categorical_columns = df.columns[df.dtypes == 'object']  
labelencoder = LabelEncoder()  
for column in categorical_columns:  
    df[column] = labelencoder.fit_transform(df[column])  
    print(dict(enumerate(labelencoder.classes_)))
```

```
{0: 'brick', 1: 'monolith', 2: 'otherwise', 3: 'panel'}  
{0: 'average', 1: 'excellent', 2: 'freelayout', 3: 'good', 4: 'requirerepair', 5: 'roughfinish'}
```

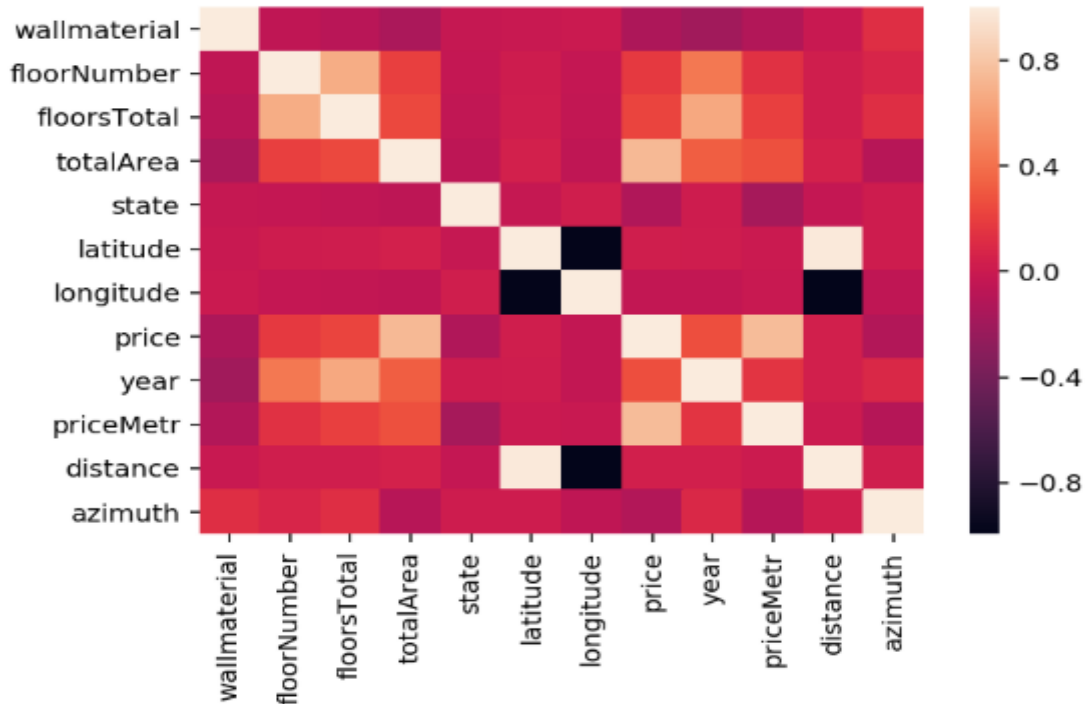
### 3.19 Сурет - категориялық параметрлерді сандыққа ауыстыру

Нәтижесінде мендегі деректердің жалпы көрінісі келесі кестеде көрсетілген.

```
df.head()
```

|   | wallmaterial | floorNumber | floorsTotal | totalArea | state | latitude  | longitude | price       | priceMetr    | distance     | azimuth |
|---|--------------|-------------|-------------|-----------|-------|-----------|-----------|-------------|--------------|--------------|---------|
| 0 | 0            | 3.0         | 5.0         | 87.2      | 3     | 43.215919 | 76.966198 | 63000000.0  | 7.224771e+05 | 3002.611574  | 145.97  |
| 1 | 0            | 2.0         | 4.0         | 58.0      | 1     | 43.218166 | 76.963467 | 54300000.0  | 9.362069e+05 | 2671.836457  | 146.90  |
| 2 | 1            | 4.0         | 7.0         | 145.0     | 1     | 43.219007 | 76.965039 | 155000000.0 | 1.068966e+06 | 2668.221347  | 143.51  |
| 3 | 0            | 1.0         | 5.0         | 60.0      | 1     | 43.265063 | 76.944800 | 27500000.0  | 4.583333e+05 | 2974.574105  | 358.96  |
| 4 | 1            | 13.0        | 13.0        | 70.0      | 4     | 43.255675 | 76.817213 | 16000000.0  | 2.285714e+05 | 10593.307394 | 280.58  |

3.20 Сурет - Алдын ала өңдеуден кейінгі деректер



3.21 Сурет - Параметрлер корреляциясы

Деректерді дайындаудың ең соңғы кезеңі ол біз болжайтын мақсатты айнымалыны таңдаймыз. Ол бізде пәтердің шаршы метрге шаққандағы бағасы.

```
y = df['priceMetr']
```

3.22 Сурет - Мақсатты айнымалыны таңдау

Ендігі кезекте модельді оқытуға қатысатын параметрлерді таңдап алып, баға көрсетілмеген жаңа X датасетін қалыптастырамыз.

```
features = [  
    'wallmaterial',  
    'floorNumber',  
    'floorsTotal',  
    'state',  
    'totalArea',  
    'year',  
    'distance',  
    'azimuth'  
]
```

```
X = df[features]
```

### 3.23 Сурет - Модельді оқыту параметрлері

Соңғы жолдағы `train_test_split()` функциясы автоматты түрде X пен y ті 4 топқа бөледі. Бұл бізге модель сапасын бейтаныс деректерге тексеруге мүмкіндік береді.

- жаттығу X;
- валидациялық (тексеру) X;
- жаттығу y;
- валидациялық y.

### 3.3 Random forest арқылы оқыту моделін құру

Бірінші модельді оқыту үшін `RandomForestRegressor` объектісін құрамыз. Оны `rf_model` ретінде сақтаймыз. Және оған бірқатар параметрлерді (яғни гиперпараметрлер), атап айтқанда:

- ағаш саны: 2000 (`n_estimators=2000`);
- ағаштың ең үлкен тереңдігі: 55 (`max_depth=55`);
- параллель іске қосылатын тапсырмалар саны: (`n_jobs=-1`);
- раскол сапасын өлшеу функциясы: (`criterion='mse'`);
- бөліністі іздеу кезінде ескеру керек функциялар саны: (`max_features=3`);
- іріктемелердің ең аз саны: (`min_samples_split=2`).

```
rf_model = RandomForestRegressor(n_estimators=2000,  
                                n_jobs=-1,  
                                bootstrap=False,  
                                criterion='mse',  
                                max_features=3,  
                                random_state=1,  
                                max_depth=55,  
                                min_samples_split=2  
                                )
```

### 3.24 Сурет - RandomForestRegressor объектісі



Құрылған модельді оқытамыз. Ол үшін fit методын қолданамыз.

```
rf_model.fit(train_X, train_y)|
RandomForestRegressor(bootstrap=False, criterion='mse', max_depth=55,
max_features=3, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=2000,
n_jobs=-1, oob_score=False, random_state=1, verbose=0,
warm_start=False)
```

### 3.25 Сурет - fit методын қолдану

Валидациялық таңдау негізінде бағаның болжамды мәнін есептейміз

```
rf_prediction = rf_model.predict(val_X).round(0)
```

```
print_metrics(rf_prediction, val_y)
```

Орташа абсолютті қате: 14.3 %

Медиандық абсолюттік қате: 7.15 %

### 3.26 Сурет - Random Forest моделінің нәтижелері

predict әдісі (val\_X) валидациялық (тексеру) X датасетінің бағасын болжайды. Ал біз жасаған print\_metrics() функциясы алдын ала болжанған және алдын ала белгілі бағаларды алады және метрика мәндерін басып шығарады.

XGBoost қарапайым кездейсоқ орманға қарағанда жоғары нәтижелерге уәде беретін градиентті бустинг кітапханасын қосамыз. Модельді құру және оқыту идеясы екі әдістеде бірдей. Біз оны белгілі гиперпараметрлермен жасаймыз, үйретеміз, болжаймыз, метрика мәндерін есептейміз.

```
xgb_model = xgb.XGBRegressor(objective = 'reg:gamma',
learning_rate = 0.01,
max_depth = 45,
n_estimators = 2000,
nthread = -1,
eval_metric = 'gamma-nloglik',
)
```

### 3.27 Сурет - XGBoostRegressor объектісі

```
: xgb_model.fit(train_X, train_y)
: XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
:   colsample_bynode=1, colsample_bytree=1,
:   eval_metric='gamma-nloglik', gamma=0, importance_type='gain',
:   learning_rate=0.01, max_delta_step=0, max_depth=45,
:   min_child_weight=1, missing=None, n_estimators=2000, n_jobs=1,
:   nthread=-1, objective='reg:gamma', random_state=0, reg_alpha=0,
:   reg_lambda=1, scale_pos_weight=1, seed=None, silent=None,
:   subsample=1, verbosity=1)
```

### 3.28 Сурет - fit методын қолдану

```
xgb_prediction = xgb_model.predict(val_X).round(0)
```

```
print_metrics(xgb_prediction, val_y)
```

Орташа абсолютті қате: 13.1 %  
Медиандық абсолюттік қате: 7.12 %

### 3.29 Сурет - XGBoostRegressor моделінің нәтежиелері

Нәтиже жақсарды. Алайда айтарлықтай жақсы көрсеткіш емес. Біздің жағдайда XGBoost орташа абсолютті қатені 1,2 %-ға, медиандық абсолюттік қатені 0,03 %-ға жақсартты.

Модельді құруда random forest құру кезінде белгілердің маңыздылығын көру жолы бар.

```

importances = rf_model.feature_importances_
std = np.std([tree.feature_importances_ for tree in rf_model.estimators_],
             axis=0)
indices = np.argsort(importances)[::-1]

print("Белгілер маңыздылық рейтингі:")
for f in range(X.shape[1]):
    print("%d. %s (%f)" % (f + 1, features[indices[f]], importances[indices[f]]))

plt.figure()
plt.title("Маңыздылық рейтингі")
plt.bar(range(X.shape[1]), importances[indices], color="b", yerr=std[indices], align="center")
plt.xticks(range(X.shape[1]), indices)
plt.xlim([-1, X.shape[1]])
plt.show()

```

```

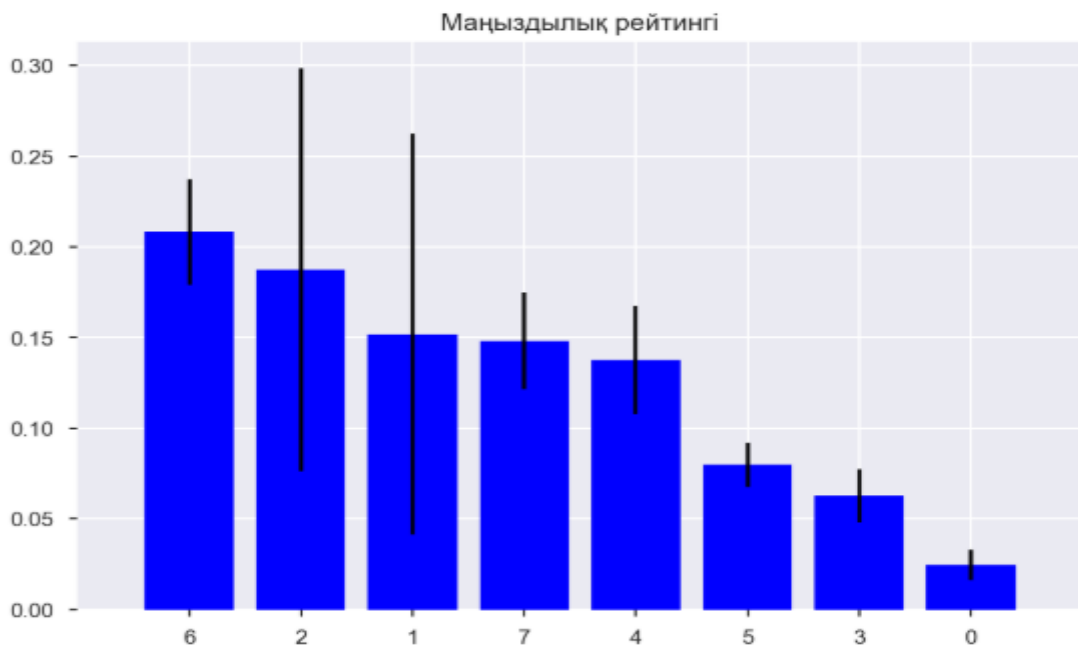
Белгілер маңыздылық рейтингі:
1. distance (0.208145)
2. floorsTotal (0.187365)
3. floorNumber (0.151959)
4. azimuth (0.148174)
5. totalArea (0.137509)
6. year (0.079775)
7. state (0.062748)
8. wallmaterial (0.024325)

```

### 3.30 Сурет - Маңыздылық рейтингісін анықтау функциясы

Белгілер маңыздылық рейтингі (2.31-сурет):

- 1) distance (0.208145);
- 2) floorsTotal (0.187365);
- 3) floor Number (0.151959);
- 4) azimuth (0.148174);
- 5) totalArea (0.137509);
- 6) year (0.079775);
- 7) state (0.062748);
- 8) wallmaterial (0.024325).



3.31 Сурет - Белгілер маңыздылық рейтингі

### 3.4 Болжам жасау процесі және нәтиже шығару

Моделді оқытқаннан кейінгі моделдің жұмысын тексереміз. Ол үшін <https://krisha.kz/> сайтынан сатуға қойылған пәтердің бағасын болжайтын боламыз.

23 970 000 ₸

Город \_\_\_\_\_ Алматы, Ауэзовский р-н  
[показать на карте](#)

Дом \_\_\_\_\_ кирпичный, 2008 г.п.

Этаж \_\_\_\_\_ 7 из 10

Площадь \_\_\_\_\_ 60 м², кухня — 15 м²

Состояние \_\_\_\_\_ евроремонт

Санузел \_\_\_\_\_ раздельный

Балкон \_\_\_\_\_ балкон

#### 3.32 Сурет - Жеке пәтердің параметрлері

Ол үшін осы пәтердің параметрлерін сипаттайтын бір жолдан датафрейм жасаймыз.

```
flat = pd.DataFrame({
    'wallmaterial':[0],
    'floorNumber':[7],
    'floorsTotal':[10],
    'state':[1],
    'totalArea':[60],
    'latitude':[43.237623],
    'longitude':[76.826600]
})
```

#### 3.33 Сурет - Датафрейм құру

Жоқ параметрлерді бізде бар параметрлер арқылы толтырып артық элементтерді датафреймнан алып тастаймыз. Ол үшін drop функциясын қолданамыз.

```

flat['distance'] = list(map(lambda x, y: geodesic(city_center_coordinates, [x, y]).meters, flat['latitude'], flat['longitude']))
flat['azimuth'] = list(map(lambda x, y: get_azimuth(x, y), flat['latitude'], flat['longitude']))
flat['distance'] = flat['distance'].round(0)
flat['azimuth'] = flat['azimuth'].round(0)

flat = flat.drop('latitude', axis=1)
flat = flat.drop('longitude', axis=1)

```

### 3.34 Сурет - Алдын ала өңдеу

Екі салынған модельдер бойынша ұсыныс бағасын болжаймыз (rf\_model және xgb\_model жетілдірілген модельдері үшін predict әдісі), нәтижені орташалаймыз және пәтер алаңына көбейтеміз. Модельмен болжанған пәтердің бағасы: 24458000 тенге

```

flat = flat.drop('latitude', axis=1)
flat = flat.drop('longitude', axis=1)

rf_prediction_flat = rf_model.predict(flat).round(0)
xgb_prediction_flat = xgb_model.predict(flat).round(0)

price = (rf_prediction_flat * 0.5 + xgb_prediction_flat * 0.5)*flat['totalArea'][0]

print(f'Модельмен болжанған пәтердің бағасы: {int(price[0].round(-3))} тенге')

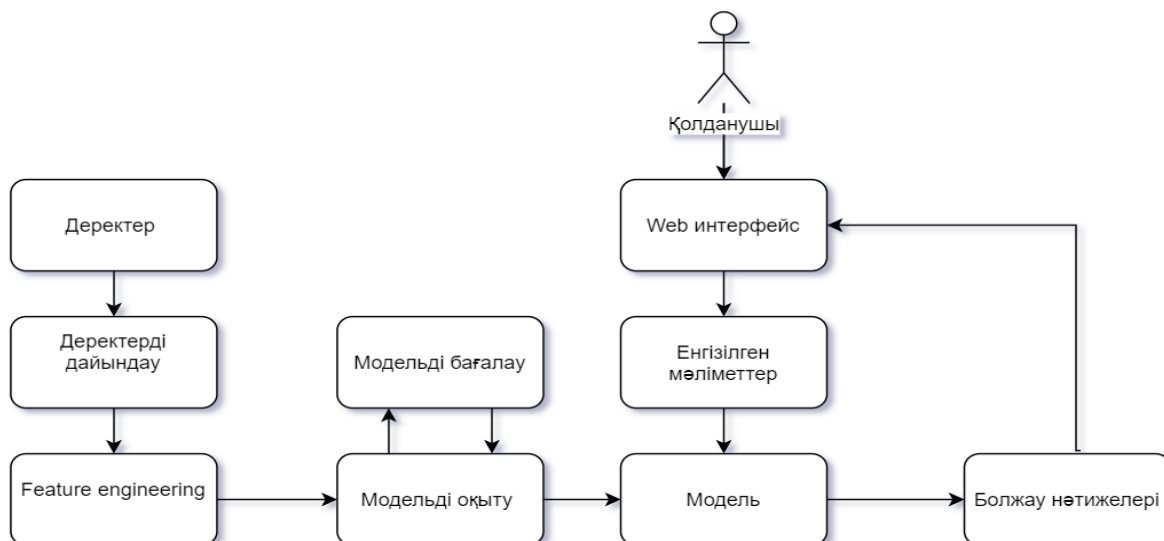
```

Модельмен болжанған пәтердің бағасы: 24458000 тенге

### 3.35 Сурет - Болжау нәтижесі

## 3.5 Веб қосымшаны әзірлеу

Машиналық оқытудың бірнеше моделдерін әзірледім, осы тестілеудің сандық болжамдарын жасап, нәтижелерін тексердім және олардың бәрін автономды түрде жасадым. Шын мәнінде, болжамдар генерациясы - бұл машиналық оқыту жобасының бір бөлігі ғана, бірақ менің ойымша, бұл ең маңызды бөлігі болды. Машиналық оқыту моделін әзірлегенде, оны қалай дамыту, яғни бұл модельді басқа пайдаланушылар үшін қалай қол жетімді ету керек деген сұраққа жауап беретін болсақ. Ол үшін веб қосымша қажет.



3.36 Сурет - Модель қолданушыларға қызмет көрсету сервисі ретінде

Web қосымша машиналық оқыту негізінде құрылған модельдің негізінде жасалған.

Модельді оқытқаннан кейін, оны қайта оқытусыз пайдалану үшін үлгіні сақтап алған болатынбыз. Мен файл түрінде модельді сақтау үшін келесі жолдарды қостым .PKL файлды одан әрі пайдалану үшін.

```

joblib.dump(rf_model, 'rf_model.pkl')
['rf_model.pkl']
  
```

3.37 Сурет - Модельді .PKL файл ретінде сақтау

```

model = open('rf_model.pkl', 'rb')
rf_model = joblib.load(model)
  
```

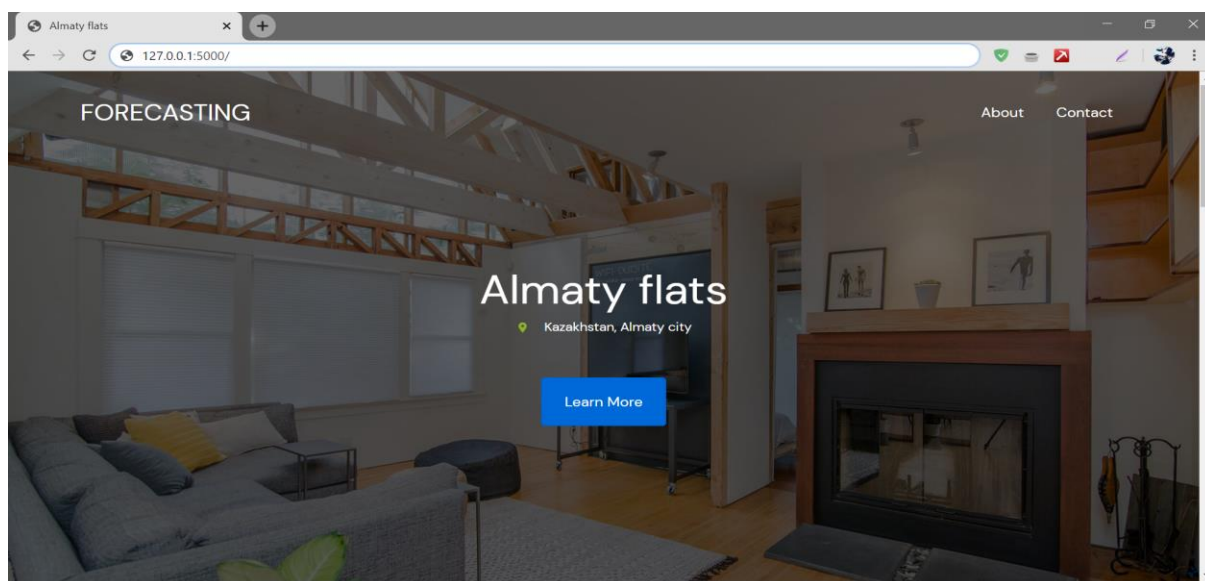
3.38 Сурет - PKL файлды веб қосымша үшін пайдалану

Мен қосымшаны бір модуль ретінде іске қостым. Ол үшін мен Flask үлгісін HTML ( templates) сол каталогта таба алатынымызды білу үшін name аргументі бар Flask жаңа экземплярын бастадым.

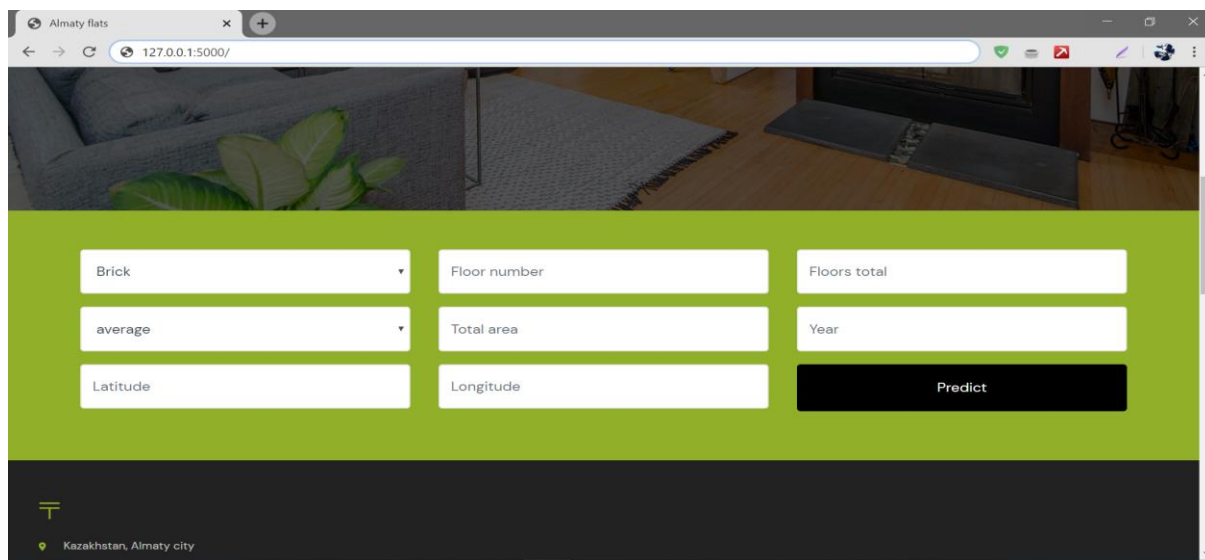
Одан әрі мен,URL мекенжайын көрсету үшін, index функциясын іске қосатын route decorator ( @app.route ( '/' )) қолдандым.

Мен Post методтын серверге деректерді беру үшін пайдаландым. Predict функция ішінде біз пәтердің бірнеше параметрлерін форма арқылы алып болжауға арналған деректер жинағына қол жеткіземіз. Біз пайдаланушы енгізген жаңа мәндерді аламыз және пәтер бағасын болжау үшін біздің модельмізді пайдаланамыз.

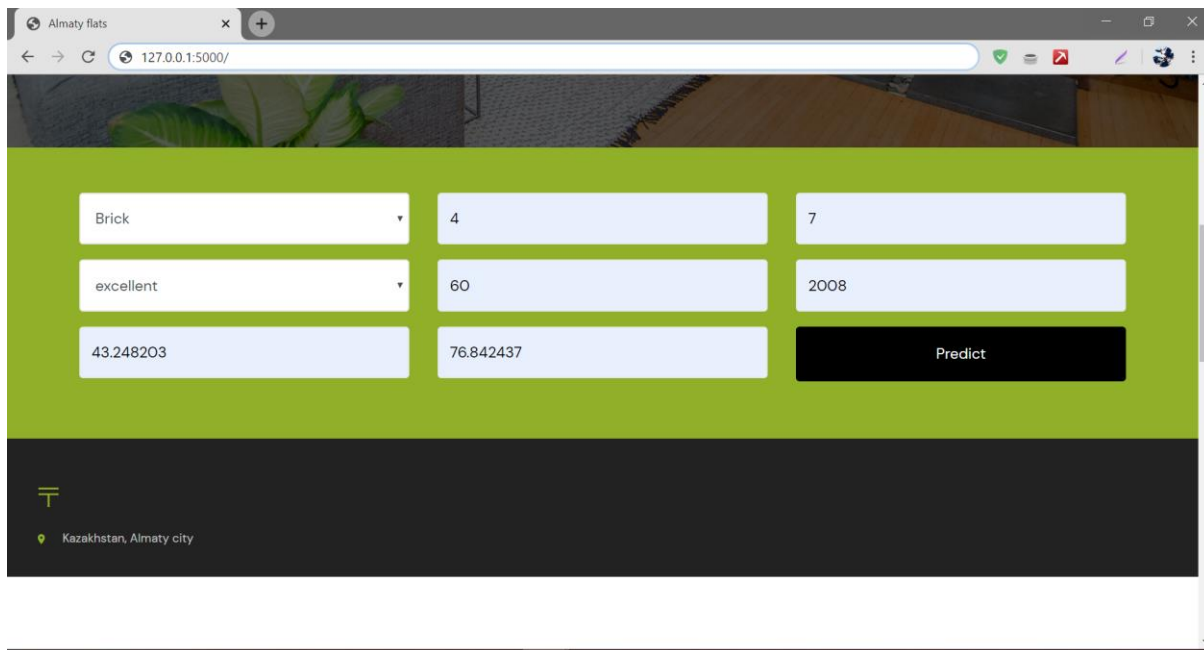
Пәтердің бағасын болжау үшін пәтердің бірнеше параметрлерін болжауға арналған форма толтырып predict батырмасын басау қажет.



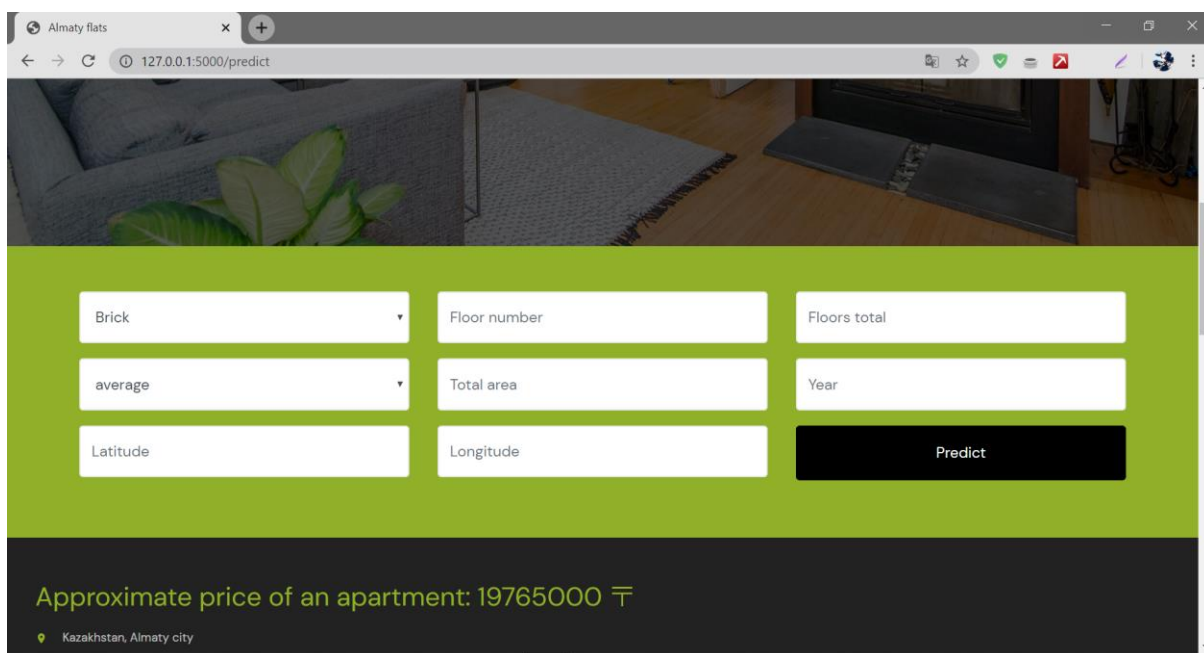
3.39 Сурет - Бастапқы бет



3.40 Сурет - Параметрлерге арналған форма



3.41 Сурет - Параметрлерді толтыру



3.42 Сурет - Болжау нәтижесін шығару



## ҚОРЫТЫНДЫ

Мен дипломдық жұмыста Алматы қаласындағы пәтер бағасын болжайтын модель жасадым. Дипломдық жұмыстың нәтижесінде машиналық оқытудың өмір сүру салалары үшін оның ішінде жылжымайтын мүлікті бағалауда пайдаланып веб қосымша әзірледім.

Жылжымайтын мүлік объектілерінің нарықтық құнын анықтау күрделі және бірегей процесс болып табылады, себебі мүлдем ұқсас екі объектіні табу мүмкін емес. Тіпті ғимараттар бір типтік жоба бойынша салынғандығымен, бірақ әртүрлі жер учаскелерінде орналасқан жағдайда олардың құны айтарлықтай өзгеше болуы мүмкін. Алайда машиналық оқытуда әртүрлі жағдайларды ескере отырып модель құрылды. Машиналық оқытуда деректер көп болған сайын болжам жасау моделінің сапасы жақсара түседі.

Мен дипломдық жұмыста көрсеткен әдістер біздің елдің бағалау тәжірибесінде әлі де аз қолданылады немесе мүлдем қолданылмайды, себебі стандарттар мен тексеруші органдар машиналық оқыту әдістерін қолданып үйренбеген. Бірақ машиналық оқыту дамып келеді, оның элементтері тек ірі технологиялық компанияларда ғана емес, шағын бизнес салаларында да қолданысқа енуде. Машиналық оқытуда қолданылатын көптеген алгоритімер бар, олар әлдеқашан ойлап табылған. Қазіргі уақытта ең бастысы сол алгоритімдерді дұрыс қолдана білу. Алгоритімдер көп болғанымен деректер аз.

## ПАЙДАЛАНЫЛГАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Matthew Kirk. Thoughtful Machine Learning with Python. - М.: О'Reilly, 2017-676.
- 2 Гифт Ной. Прагматичный ИИ Машинное обучение и облачные технологии. - . М.: Питер, 2019 - 254 б.
- 3 David Paper. Data Science Fundamentals for Python and MongoDB. - М.: Apress, 2018 - 182 б.
- 4 Крис Элбон. Машинное обучение с использованием Python. -М.: О'Reilly, 2019 - 110 б.
- 5 Марк Лутц. Карманный справочник Python. - М.: Диалектика, 2019 - 320 б.
- 6 Sayan Mukhopadhyay. Advanced Data Analytics Using Python. -М.: Apress, 2018 - 49 б.
- 7 Flask framework [Электронды ресурс]. - Ресурсқа кіру: <https://flask.palletsprojects.com/en/1.1.x/>
- 8 Введение в машинное обучение с помощью scikit-learn [Электронды ресурс]. - Ресурсқа кіру: <https://habr.com/ru/post/264241/>
- 9 NumPy в Python [Электронды ресурс]. - Ресурсқа кіру: <https://habr.com/ru/post/352678/>
- 10 Введение в pandas: анализ данных на Python [Электронды ресурс]. - Ресурсқа кіру: <https://khashtamov.com/ru/pandas-introduction/>
- 11 Understanding Random Forest [Электронды ресурс]. - Ресурсқа кіру: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

## А қосымшасы

app.py листингі

```
import joblib
import numpy as np
import pandas as pd
from geopy.distance import geodesic
import math
from flask import Flask, request, jsonify, render_template
import pickle
import warnings
warnings.filterwarnings('ignore')
app = Flask(__name__)
model = open('rf_model.pkl', 'rb')
rf_model = joblib.load(model)
def get_azimuth(latitude, longitude):
    rad = 6372795
    city_center_coordinates = [43.238293, 76.945465]
    llat1 = city_center_coordinates[0]
    llong1 = city_center_coordinates[1]
    llat2 = float(latitude)
    llong2 = float(longitude)
    lat1 = llat1 * math.pi / 180.
    lat2 = llat2 * math.pi / 180.
    long1 = llong1 * math.pi / 180.
    long2 = llong2 * math.pi / 180.
    c11 = math.cos(lat1)
    c12 = math.cos(lat2)
    s11 = math.sin(lat1)
    s12 = math.sin(lat2)
    delta = long2 - long1
    cdelta = math.cos(delta)
    sdelta = math.sin(delta)
    y = math.sqrt(math.pow(c12 * sdelta, 2) + math.pow(c11 * s12 - s11 * c12 * cdelta, 2))
    x = s11 * s12 + c11 * c12 * cdelta
    ad = math.atan2(y, x)

    x = (c11 * s12) - (s11 * c12 * cdelta)
    y = sdelta * c12
    z = math.degrees(math.atan(-y / x))
    if (x < 0):
        z = z + 180.
    z2 = (z + 180.) % 360. - 180.
```

## А ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
z2 = - math.radians(z2)
    anglerad2 = z2 - ((2 * math.pi) * math.floor((z2 / (2 * math.pi))))
    angledeg = (anglerad2 * 180.) / math.pi
    return round(angledeg, 2)
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict',methods=['POST'])
def predict():
    int_features = [float(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    wallmaterial = request.form.get('wallmaterial')
    floorNumber = request.form.get('floorNumber')
    floorsTotal = request.form.get('floorsTotal')
    state = request.form.get('state')
    totalArea = request.form.get('totalArea')
    year = request.form.get('year')
    latitude = request.form.get('latitude')
    longitude = request.form.get('longitude')
    flat = pd.DataFrame({
        'wallmaterial': [wallmaterial],
        'floorNumber': [floorNumber],
        'floorsTotal': [floorsTotal],
        'state': [state],
        'totalArea': [totalArea],
        'year': [year],
        'latitude': [latitude],
        'longitude': [longitude]
    })
    city_center_coordinates = [43.238293, 76.945465]
    flat['distance'] = list(map(lambda x, y: geodesic(city_center_coordinates, [x,
y]).meters, flat['latitude'], flat['longitude']))
    flat['azimuth'] = list(map(lambda x, y: get_azimuth(x, y), flat['latitude'],
flat['longitude']))
    flat['distance'] = flat['distance'].astype(float)
    flat['azimuth'] = flat['azimuth'].astype(float)
    flat['distance'] = flat['distance'].round(0)
    flat['azimuth'] = flat['azimuth'].round(0)
    flat = flat.drop('latitude', axis=1)
    flat = flat.drop('longitude', axis=1)
    prediction = rf_model.predict(flat).round(0)
    flat['totalArea'] = flat['totalArea'].astype(float)
```

## А қосымшасының жалғасы

```
price = prediction * flat['totalArea']
    output = int(price[0].round(-3))
    return render_template('index.html', prediction_text='Approximate price of an
apartment: {}'.format(output))
@app.route('/results',methods=['POST'])
def results():
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])
    output = prediction[0]
    return jsonify(output)
if __name__ == "__main__":
    app.run(debug=True)
```

index.html листингі

```
<!DOCTYPE html>
<html>
<head>
    <title>Almaty flats</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <link
                                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
    integrity="sha384-
Vko08x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifj
h" crossorigin="anonymous">
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
    integrity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
    crossorigin="anonymous"></script>
    <script src="https://api-maps.yandex.ru/2.1/?lang=ru_RU&apikey=<f3fe04e3-
2e96-4c00-acd8-6d8a1d0825d2>"
    type="text/javascript"></script>
    <script src="reverse_geocode.js" type="text/javascript"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <link
href="https://fonts.googleapis.com/css?family=DM+Sans:300,400,700&display=swa
p" rel="stylesheet">
    <link rel="stylesheet" href="static/fonts/icomoon/style.css">
    <link rel="stylesheet" href="static/css/bootstrap.min.css">
    <link rel="stylesheet" href="static/css/bootstrap-datepicker.css">
```

## А ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
<link rel="stylesheet" href="static/css/jquery.fancybox.min.css">
  <link rel="stylesheet" href="static/css/owl.carousel.min.css">
  <link rel="stylesheet" href="static/css/owl.theme.default.min.css">
  <link rel="stylesheet" href="static/fonts/flaticon/font/flaticon.css">
  <link rel="stylesheet" href="static/css/aos.css">
  <!-- MAIN CSS -->
  <link rel="stylesheet" href="static/css/style.css">
</head>
<body data-spy="scroll" data-target=".site-navbar-target" data-offset="300">
<div class="login">
  <!-- Main Input For Receiving Query to our ML -->
  <div class="site-wrap" id="home-section">
    <div class="site-mobile-menu site-navbar-target">
      <div class="site-mobile-menu-header">
        <div class="site-mobile-menu-close mt-3">
          <span class="icon-close2 js-menu-toggle"></span>
        </div>
      </div>
      <div class="site-mobile-menu-body"></div>
    </div>
    <header class="site-navbar site-navbar-target" role="banner">
      <div class="container">
        <div class="row align-items-center position-relative">
          <div class="col-3 ">
            <div class="site-logo">
              <a href="index.html">Forecasting</a>
            </div>
          </div>
          <div class="col-9 text-right">
            <span class="d-inline-block d-lg-none"><a href="#"
              class="text-white site-menu-toggle js-menu-
toggle py-5 text-white"><span
              class="icon-menu h3 text-white"></span></a></span>
            <nav class="site-navigation text-right ml-auto d-none d-lg-block"
role="navigation">
              <ul class="site-menu main-menu js-clone-nav ml-auto ">
                <li><a href="about.html" class="nav-link">About</a></li>
                <li><a href="contact.html" class="nav-link">Contact</a></li>
              </ul>
            </nav>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

## А ҚОСЫМШАСЫНЫҢ ЖАЛҒАСЫ

```
</div>
</header>
<div class="ftco-blocks-cover-1">
  <div class="site-section-cover overlay" data-stellar-background-ratio="0.5"
    style="background-image: url('static/images/hero_1.jpg')">
    <div class="container">
      <div class="row align-items-center justify-content-center text-center">
        <div class="col-md-7">
          <h1 class="mb-2">Almaty flats</h1>
          <p class="text-center mb-5"><span
            class="small address d-flex align-items-center justify-content-
center"> <span
              class="icon-room mr-3 text-primary"></span>
<span>Kazakhstan, Almaty city</span></span>
          </p>
          <p><a href="#" class="btn btn-primary text-white px-4 py-3">Learn
More</a></p>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="realestate-tabpane pb-5" style="padding-top: 48px;">
  <div class="container tab-content">
    <div class="tab-pane active" id="for-rent" role="tabpanel" aria-
labelledby="rent-tab">
      <div class="row">
        <div class="col-md-4 form-group">
          <select name="wallmaterial" class="form-control w-100"
required="required">
            <option value="0">Brick</option>
            <option value="1">Monolith</option>
            <option value="2">Otherwise</option>
            <option value="3">Panel</option>
          </select>
        </div>
        <div class="col-md-4 form-group">
          <input type="text" name="floorNumber" class="form-control"
placeholder="Floor number">

```

## А қосымшасының жалғасы

```
</div>
  <div class="col-md-4 form-group">
    <input type="text" name="floorsTotal" class="form-control"
placeholder="Floors total">
  </div>
</div>

<div class="row">
  <div class="col-md-4 form-group">
    <select name="state" class="form-control w-100"
placeholder="State">
      <option value="0">average</option>
      <option value="1">excellent</option>
      <option value="2">freelayout</option>
      <option value="3">good</option>
      <option value="4">requirerepair</option>
      <option value="5">roughfinish</option>
    </select>
  </div>
  <div class="col-md-4 form-group">
    <input type="text" name="totalArea" class="form-control"
placeholder="Total area">
  </div>
  <div class="col-md-4 form-group">
    <input type="text" name="year" class="form-control"
placeholder="Year">
  </div>
  <div class="col-md-4 form-group">
    <input type="text" name="latitude" class="form-control"
placeholder="Latitude">
  </div>
  <div class="col-md-4 form-group">
    <input type="text" name="longitude" class="form-control"
placeholder="Longitude">
  </div>
  <div class="col-md-4">
    <input type="submit" class="btn btn-black py-3 btn-block"
value="Predict">
  </div>
</div>
</div>
</div>
```





## А қосымшасының жалғасы

```
<script src="js/popper.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/owl.carousel.min.js"></script>
<script src="js/jquery.sticky.js"></script>
<script src="js/jquery.waypoints.min.js"></script>
<script src="js/jquery.animateNumber.min.js"></script>
<script src="js/jquery.fancybox.min.js"></script>
<script src="js/jquery.stellar.min.js"></script>
<script src="js/jquery.easing.1.3.js"></script>
<script src="js/bootstrap-datepicker.min.js"></script>
<script src="js/aos.js"></script>
<script src="js/main.js"></script>
</div>
<div id="map"></div>
</body>
</html>
```