

УНИВЕРЗИТЕТ У НИШУ

Електронски факултет

Извештај

Паметни систем за праћење и
оптимизацију буџета

Студенти:

Бојан Митић 1918

Вукашин Поповић 1927

Ментор:

Проф. др. Леонид Стоименов

Contents

1.	Увод	3
2.	Опис проблема	4
3.	Коришћене технологије	5
3.1.	Python.....	5
3.2.	FastAPI	5
3.3.	Pandas i NumPy	6
3.4.	Matplotlib	6
3.5.	TensorFlow i Keras.....	6
4.	Вештачка интелигенција	7
4.1.	Машинско учење	7
4.2.	Неуронске мреже.....	8
4.3.	Рекурентне неуронске мреже	8
5.	LSTM – Long Short-Term Memory	8
5.1.	Структура LSTM ћелије	9
6.	Суп података	10
6.1.	Структура CSV фајла	11
6.2.	Обрада података.....	11
7.	Архитектура система	11
7.1.	Почетна архитектура (један модел)	12
7.2.	Еволуција архитектуре (три модел)	12
7.3.	Експеримент са више модела (седам модел)	13
8.	Rolling walk-forward симулација	14
9.	Window size и избор параметара	16
9.1.	Window size (величина временског прозора)	16
9.2.	Избор прага тачности (accuracy threshold)	17
9.3.	Избор и улога скалера (StandardScaler)	17
10.	API функционалности.....	17
10.1.	/simulate.....	18
10.2.	/simulateTomorrow.....	19
11.	Ограничења	20
12.	Могућа унапређења.....	21
13.	Закључак	22
14.	Литература	24

1. Увод

Вештачка интелигенција представља једну од најзначајнијих области савремене информатике и има све већу улогу у решавању комплексних проблема у различитим сферама друштва. Захваљујући способности да анализира велике количине података, препознаје обрасце и доноси закључке на основу претходних искустава, вештачка интелигенција данас налази примену у медицини, финансијама, индустрији, саобраћају, као и у свакодневном животу појединаца. Њена основна вредност огледа се у могућности аутоматизације процеса и подршке доношењу одлука које су засноване на подацима, а не искључиво на интуицији.

Једна од посебно значајних примена вештачке интелигенције јесте анализа временских серија и предвиђање будућих вредности на основу историјских података. Овај приступ је изузетно користан у области личних финансија, где корисници често немају јасан увид у структуру и динамику својих месечних трошкова. Коришћењем напредних модела машинског учења могуће је предвидети будуће издатке, уочити трендове потрошње и на тај начин омогућити боље планирање буџета.

Оптимизација месечних трошкова уз помоћ предиктивних модела заснованих на вештачкој интелигенцији представља корак ка паметнијем управљању личним финансијама. Уместо реактивног приступа, где се финансијске одлуке доносе након што трошкови већ настану, предвиђање омогућава проактивно деловање и благовремене корекције навика потрошње. Управо из тог разлога, развој апликација које користе вештачку интелигенцију за анализу и предвиђање финансијских података представља актуелан и практично значајан проблем, који ће бити предмет овог рада.

2. Опис проблема

Основни циљ овог пројекта јесте предвиђање дневних трошкова корисника на основу историјских финансијских података, са намером да се омогући боље планирање и разумевање потенцијалних будућих расхода. Анализом претходних трошкова и препознавањем понављајућих образаца у потрошњи, кориснику се пружа увид у очекиване издатке за наредне дане, што може значајно допринети рационалнијем управљању личним финансијама.

Међутим, предвиђање финансијских трошкова представља сложен проблем, пре свега због природе самих података. Дневни трошкови су подложни великом броју спољашњих и унутрашњих фактора као што су непланиране куповине, ванредни догађаји, промене животних навика, сезонске осцилације, као и психолошки фактори који утичу на понашање корисника. Ови фактори често нису експлицитно присутни у подацима и не могу се директно моделирати, што уноси значајну дозу неизвесности у процес предвиђања.

Додатни проблем представља чињеница да финансијски подаци често садрже шум, неправилности и неуједначену учесталост трошкова. На пример, поједини дани могу имати више трансакција различитих износа, док други дани могу бити без икаквих расхода. Оваква неуједначеност отежава изградњу стабилног модела који би једнако добро функционисао у свим ситуацијама. Такође, ограничена количина историјских података, нарочито код нових корисника, додатно смањује поузданост предвиђања.

Иако напредни модели вештачке интелигенције, као што су рекурентне неуронске мреже, могу успешно да науче сложене временске зависности у подацима, неопходно је нагласити да је потпуно тачно предвиђање трошкова немогуће у 100% случајева. Предиктивни модел у овом контексту не треба посматрати као средство за апсолутно тачну прогнозу, већ као алат за процену вероватних сценарија и идентификацију трендова у потрошњи.

Стога, проблем који овај пројекат адресира није елиминација неизвесности у финансијском понашању, већ њено смањење кроз аналитички приступ и коришћење историјских података. Циљ је да се кориснику обезбеди информисана основа за доношење одлука, уз јасно разумевање да свака прогноза представља апроксимацију реалног стања, а не његову потпуну репрезентацију.

3. Коришћене технологије

За реализацију апликације за предвиђање дневних трошкова коришћен је скуп савремених софтверских технологија које омогућавају ефикасну обраду података, развој модела вештачке интелигенције, као и њихову интеграцију у функционалан серверски систем. Комбинација програмског језика Python, библиотека за анализу података и оквира за машинско учење омогућила је изградњу флексибилне и прошириве архитектуре апликације.

3.1. Python

Python је интерпретирани програмски језик високог нивоа који се широко користи у области анализе података, машинског учења и вештачке интелигенције. Његова једноставна синтакса, велика заједница корисника и богат екосистем библиотека чине га идеалним избором за развој апликација заснованих на подацима.

У овом пројекту Python је коришћен као главни програмски језик за имплементацију логике апликације, обраду улазних података, управљање моделима машинског учења, као и за креирање REST API сервиса. Захваљујући својој флексибилности, Python омогућава брзо прототипирање и једноставну модификацију архитектуре, што је било посебно важно током експериментисања са различитим моделима предвиђања.

3.2. FastAPI

FastAPI је модеран Python оквир за развој веб API сервиса, заснован на асинхроним механизмима и стандардима као што су OpenAPI и JSON Schema. Карактерише га висока перформанса, једноставна дефиниција рута и аутоматско генерисање документације.

У оквиру овог пројекта, FastAPI је коришћен за имплементацију серверске логике која омогућава комуникацију између клијентске апликације и модела машинског учења. Путем FastAPI рута омогућено је покретање симулација, добијање предвиђања за наредни дан, као и враћање резултата у облику JSON података или графичких приказа. Аутоматски генерисана Swagger документација знатно је олакшала тестирање и развој API функција.

3.3. Pandas i NumPy

Pandas и NumPy су основне Python библиотеке за рад са подацима и нумеричке прорачуне. NumPy обезбеђује ефикасне структуре података за рад са вишедимензионалним низовима и математичким операцијама, док Pandas нуди напредне алате за манипулацију табеларним подацима.

У овој апликацији Pandas је коришћен за учитавање и обраду финансијских података из CSV фајлова, филтрирање трансакција, као и агрегацију више дневних трошкова у јединствену дневну суму. NumPy је коришћен за припрему података за улаз у LSTM моделе, рад са временским серијама и израчунавање различитих статистичких мера које се користе у евалуацији модела.

3.4. Matplotlib

Matplotlib је библиотека за визуализацију података у Python-у која омогућава креирање различитих типова графикана, као што су линијски дијаграми, хистограми и површински графици.

У пројекту је Matplotlib коришћен за приказ резултата симулација и поређење стварних и предвиђених вредности дневних трошкова. Генерисани графици омогућавају визуелну анализу тачности модела, као и уочавање трендова и одступања у предвиђањима. Ови графички прикази могу бити коришћени у документацији или у оквиру корисничког интерфејса апликације.

3.5. TensorFlow i Keras

TensorFlow је оквир за машинско учење и дубоке неуронске мреже, док Keras представља виши ниво апстракције који омогућава једноставније дефинисање, тренирање и коришћење модела.

У овом пројекту TensorFlow и Keras су коришћени за имплементацију LSTM (Long Short-Term Memory) неуронских мрежа, које су посебно погодне за рад са временским серијама. Модел је трениран да на основу историјских података о дневним трошковима предвиђају будуће вредности. Током развоја система експериментисано је са различитим архитектурама, укључујући један модел, затим више специјализованих модела, што је омогућило боље прилагођавање различитим нивоима потрошње.

4. Вештачка интелигенција

Вештачка интелигенција (ВИ) представља област рачунарских наука која се бави развојем система способних да извршавају задатке који иначе захтевају људску интелигенцију. Такви задаци обухватају учење, закључивање, препознавање образаца, доношење одлука и предвиђање будућих догађаја на основу постојећих података.

Основна идеја вештачке интелигенције јесте да се рачунарским системима омогући да на основу искуства, односно података, унапређују своје перформансе без експлицитног програмирања сваког корака. У савременим апликацијама, ВИ се користи у широком спектру области као што су финансије, медицина, саобраћај, индустрија и личне финансије, где анализа великих количина података може довести до бољих и информисанијих одлука.

У контексту овог пројекта, вештачка интелигенција се користи као средство за анализу историјских финансијских података и предвиђање будућих трошкова, са циљем да се кориснику омогући боље планирање и оптимизација личне потрошње.

4.1. Машинско учење

Машинско учење представља подобласт вештачке интелигенције која се фокусира на развој алгоритама који омогућавају системима да уче из података и унапређују своје резултате током времена. За разлику од класичних програмских приступа, где су правила унапред дефинисана, код машинског учења модел самостално проналази обрасце у подацима.

Постоје три основне категорије машинског учења: надгледано учење, ненадгледано учење и учење уз појачање. У овом пројекту коришћено је надгледано учење, где модел учи на основу историјских података који садрже улазне вредности (дневне трошкове) и познате излазне вредности (стварне трошкове наредног дана).

Машинско учење је посебно погодно за финансијске примене, јер омогућава обраду великих количина података и прилагођавање променљивим обрасцима потрошње, који се временом мењају у зависности од навика корисника.

4.2. Неуронске мреже

Неуронске мреже су рачунарски модели инспирисани начином функционисања људског мозга. Састоје се од вештачких неурона који су међусобно повезани и организовани у слојеве: улазни, један или више скривених слојева и излазни слој.

Сваки неурон прима улазне сигнале, обрађује их помоћу тежинских коефицијената и активационе функције, и прослеђује резултат следећем слоју. Током процеса тренирања, неуронска мрежа прилагођава своје тежине тако да минимизује грешку између предвиђених и стварних вредности.

Због своје способности да моделирају сложене нелинеарне односе, неуронске мреже су посебно ефикасне у задацима као што су анализа временских серија, препознавање образаца и предвиђање будућих вредности, што их чини погодним избором за предвиђање финансијских трошкова.

4.3. Рекурентне неуронске мреже

Рекурентне неуронске мреже (RNN – Recurrent Neural Networks) представљају посебну врсту неуронских мрежа које су дизајниране за рад са секвенцијалним подацима и временским серијама. За разлику од класичних неуронских мрежа, RNN модели имају повратне везе које им омогућавају да задрже информације о претходним улазима.

Ова особина је изузетно значајна код проблема где редослед података игра кључну улогу, као што су финансијски подаци, где тренутна потрошња често зависи од потрошње у претходним данима. Међутим, класичне рекурентне мреже имају ограничења у виду проблема нестајања и експлозије градијента, што отежава учење на дужим секвенцама.

Због наведених ограничења, у пракси се често користе напредније варијанте рекурентних мрежа, као што су LSTM мреже, које су посебно погодне за анализу дугорочних зависности у временским серијама. Управо ова врста мреже представља основу модела коришћених у овом пројекту.

5. LSTM – Long Short-Term Memory

LSTM (Long Short-Term Memory) представља напредну архитектуру рекурентних неуронских мрежа, дизајнирану са циљем да превазиђе ограничења класичних RNN модела, пре свега проблем нестајања и експлозије градијента. Овај проблем се јавља приликом учења на дужим временским секвенцама, где модел губи способност да „памти“ информације које су се догодиле у даљој прошлости.

Основна предност LSTM мреже лежи у њеној способности да задржи релевантне информације током дужег временског периода, док истовремено одбацује небитне податке. Ова карактеристика чини LSTM изузетно погодним за анализу временских серија, као што су финансијски подаци, где постоје сложене и дугорочне зависности између претходних и будућих вредности.

У оквиру овог пројекта, LSTM модели се користе за предвиђање дневних трошкова на основу историјских података, при чему модел учи обрасце потрошње који се понављају током времена, као што су периодични трошкови, нагле промене или трендови.

5.1. Структура LSTM ћелије

LSTM ћелија представља основни градивни елемент LSTM мреже и састоји се од више компоненти које заједно омогућавају контролисано памћење и заборављање информација. За разлику од класичног неурона, LSTM ћелија садржи унутрашње стање (енг. cell state) које служи као носилац дугорочне меморије.

Кључне компоненте LSTM ћелије су:

- Стање ћелије (Cell State)

Стање ћелије представља главни канал кроз који информације пролазе кроз време. Оно омогућава LSTM мрежи да задржи податке током више временских корака уз минималне измене, чиме се омогућава учење дугорочних зависности.

- Врата заборава (Forget Gate)

Врата заборава одлучују које информације из претходног стања ћелије треба задржати, а које одбацити. Овај механизам је од изузетне важности јер омогућава моделу да елиминише застареле или небитне податке, као што су једнократни трошкови који не утичу на будуће обрасце потрошње.

- Улазна врата (Input Gate)

Улазна врата контролишу које нове информације ће бити додате у стање ћелије. Овај део LSTM ћелије омогућава моделу да научи које тренутне вредности су значајне за будућа предвиђања.

- Кандидатско стање (Candidate State)

Кандидатско стање представља нове потенцијалне информације које могу бити уписане у стање ћелије. У комбинацији са улазним вратима, одређује се у којој мери ће ове информације утицати на меморију.

- Излазна врата (Output Gate)

Излазна врата дефинишу који део стања ћелије ће бити прослеђен као излаз из LSTM ћелије. На овај начин, модел контролише које информације ће утицати на наредне слојеве или на финалну прогнозу.

Комбинацијом наведених механизма, LSTM ћелија постиже баланс између памћења и заборављања, што је од кључне важности за рад са временским серијама. У практичној примени овог пројекта, LSTM структура омогућава моделу да користи информације из претходних дана како би што прецизније предвидео будуће дневне трошкове, узимајући у обзир како краткорочне, тако и дугорочне трендове.

6. Суп података

За израду и тестирање апликације коришћен је скуп података који представља историјске финансијске трансакције. Податке је потребно прикупити у CSV формату, где сваки ред представља једну трансакцију.

6.1. Структура CSV фајла

Скуп података садржи барем следеће колоне:

Колона	Тип података	Опис
Date	датум (YYYY-MM-DD)	Датум обављене трансакције
Amount	број (float)	Вредност трошка или прихода
Type	текст (string)	"Expense" или "Income"

Пример неколико редова из фајла:

Date	Amount	Type
2025-01-01	45.50	Expense
2025-01-01	120.00	Expense
2025-01-02	200.00	Income
2025-01-02	30.75	Expense

Важно је напоменути да апликација тренутно користи само трошкове (Type == "Expense") за предвиђање будућих расхода.

6.2. Обрада података

7. Архитектура система

Архитектура апликације је осмишљена тако да омогући предвиђање дневних трошкова корисника на основу историјских података. Главни фокус је био на коришћењу LSTM модела за временске серије и организацији апликације тако да буде лако проширива и одржива.

7.1. Почетна архитектура (један модел)

У почетној верзији апликације коришћен је један LSTM модел за предвиђање свих будућих дневних трошкова. Основне карактеристике ове архитектуре биле су:

- Један универзални модел који је трениран на целокупним подацима.
- Улаз у модел је био низ претходних WINDOW_SIZE дана (нпр. 7 или 11) са скалираним вредностима трошкова.
- Излаз је био предвиђени трошак за следећи дан.
- Апликација је обрађивала дневне серије и враћала прогнозу кроз REST API (FastAPI).

Недостаци ове архитектуре:

- Један модел није могао добро да предвиђа различите категорије потрошње (мале, средње, велике трошкове).
- У неким случајевима предвиђања су била превисока или прениска због високих варијација у подацима.
- Тачност је била ограничена, поготово у данима са необичним трошковима.
- Ту можеш да убациш скриншот или шему почетне архитектуре апликације.

7.2. Еволуција архитектуре (три модел)

Након почетних тестова, архитектура је ревидирана на три LSTM модела, подељена према нивоу потрошње:

- Модел "low" – за предвиђање малих трошкова (до 33% квантиле).
- Модел "mid" – за предвиђање средњих трошкова (између 33% и 66% квантиле).
- Модел "high" – за предвиђање великих трошкова (изнад 66% квантиле).

Карактеристике нове архитектуре:

- Изабран модел се динамички бира на основу последње стварне потрошње.
- Сваки модел је независно трениран на делу података који одговара његовој категорији.
- Овај приступ повећава прецизност, јер сваки модел учи специфичне обрасце за свој сегмент потрошње.
- Омогућава боље руковање екстремним вредностима и смањује грешке у предвиђањима.

Предности три модела у односу на један:

- Побољшана тачност предвиђања.
- Мање "прекомерно" или "недовољно" предвиђање у екстремним данима.
- Модуларност – могуће је накнадно додати нове категорије или прилагодити постојеће моделе.
- Овде можеш да убациш шему са три модела и да објасниш како подаци пролазе кроз одговарајући модел.

7.3. Експеримент са више модела (седам модел)

Поред три модела, извршени су експерименти са седам независних LSTM модела, где је сваки модел покушавао да предвиди тачно један сегмент потрошње:

Циљ је био да се боље разликују фини сегменти потрошње (врло ниски, ниски, до средњих, средњи, високи, веома високи, екстремни трошкови).

Тестирање је показало да повећање броја модела на седам није значајно побољшало прецизност у поређењу са три модела.

Сложеност система се драстично повећала, а време предвиђања и одржавање постали су тежи.

На основу резултата, архитектура са три модела је изабрана као оптимална комбинација тачности и једноставности.

Овде можеш да убациш графиконе или табеле са поређењем тачности 1, 3 и 7 модела.

8. Rolling walk-forward симулација

Rolling walk-forward симулација представља технику која се често користи у анализи временских серија, нарочито у финансијама, машинском учењу и предвиђању трошкова. Основна идеја ове методе је да се модел тренутно тренира и тестира на "клизном прозору" историјских података, што омогућава евалуацију његове способности да предвиђа будуће вредности у реалним условима.

Принцип рада rolling walk-forward симулације може се описати у неколико корака:

- Подела података на прозоре (windows)

Историјски подаци се деле на временске прозоре фиксне дужине (WINDOW_SIZE), који представљају број претходних дана који ће се користити за предвиђање следећег дана. На пример, ако је WINDOW_SIZE = 11, модел ће за сваки дан користити претходних 11 дана да предвиди трошак за сутра.

- Клизни прозор (rolling)

Прозор се "клизне" за један дан напред након сваке прогнозе. На тај начин се симулација изводи корак по корак, а модел увек користи најсвежије податке.

- Feed-back реалних података

Након што се предвиђање за одређени дан добије, реални трошак тог дана се користи као део прозора за следећу прогнозу. Ово омогућава да модел увек има на располагању стварне резултате из претходних дана, чиме се смањује акумулирана грешка.

- Евалуација тачности

Сваки корак предвиђања се упоређује са стварним трошком. У мом пројекту користи се $\pm 30\%$ као праг тачности, што омогућава идентификацију дана када модел адекватно предвиди потрошњу и дана када је грешка превелика.

Примена и решавање проблема:

У контексту апликације за предвиђање дневних трошкова, rolling walk-forward симулација решава неколико кључних проблема:

- Динамичка промена потрошње

Потрошња корисника није константна и може значајно варирати од дана до дана. Клизни прозор омогућава да се модел прилагођава најновијим обрасцима у подацима.

- Реалистична процена будућих трошкова

Пошто модел увек користи стварне податке из претходних дана, предвиђања за сутрашњи дан или наредне дане одражавају актуелне трендове и обрасце потрошње.

- Контрола тачности предвиђања

Walk-forward симулација омогућава евалуацију модела на сваком кораку и даје прецизне метрике као што су MAE, RMSE, R^2 и проценат предвиђања унутар задатог прага. Ове информације могу бити корисне за корисника који жели да оптимизује месечни буџет.

- Модуларност и тестирање различитих модела

Методологија клизног прозора је посебно корисна када се користе више модела (као у овом пројекту са три LSTM модела), јер омогућава динамички избор модела за различите нивое потрошње на основу последњег стварног дана.

Предности за мој пројекат:

- Омогућава прецизније дневно предвиђање трошкова у реалном времену.
- Смањује ризик од акумулирања грешке у дугим низовима предвиђања.
- Дозвољава лакше визуелно праћење помоћу графика (Matplotlib) који приказују предвиђене и стварне трошкове, са ознакама тачности.
- Флексибилна је за додавање нових категорија потрошње или прилагођавање прозора.

9. Window size и избор параметара

Један од кључних фактора који директно утичу на квалитет предвиђања у моделима за временске серије јесте правилан избор параметара. У овом пројекту посебан значај има параметар window size, као и пратећи параметри који дефинишу понашање LSTM модела и rolling walk-forward симулације.

9.1. Window size (величина временског прозора)

Window size представља број претходних временских корака (дана) који се користе као улаз у модел за предвиђање наредног дана. У контексту овог пројекта, то значи да се дневни трошкови из претходних N дана користе за прогнозу трошка за сутрашњи дан.

На пример:

- Ако је window size = 7, модел користи податке за последњих седам дана.
- Ако је window size = 11, модел користи податке за последњих једанаест дана.

У имплементацији пројекта изабрана је вредност WINDOW_SIZE = 11, као компромис између краткорочних и дугорочних образаца потрошње.

Избор window size параметра има следеће последице:

- Премали прозор

Модел реагује само на краткорочне промене и постаје осетљив на случајне осцилације у потрошњи. То може довести до нестабилних и нетачних предвиђања.

- Превелики прозор

Модел добија више историјских података, али ризикује да изгуби осетљивост на актуелне трендове. Поред тога, већи прозори повећавају сложеност модела и могу довести до прекомерног прилагођавања (overfitting).

Због тога је избор вредности 11 дана оцењен као оптималан за конкретан скуп података и навике потрошње корисника.

9.2. Избор прага тачности (accuracy threshold)

У пројекту се користи праг тачности од $\pm 30\%$, што значи да се предвиђање сматра тачним уколико се предвиђени износ налази унутар 30% од стварне вредности.

Овај приступ је изабран из практичних разлога:

- Дневни трошкови могу значајно варирати.
- Потпуна прецизност (100%) није реално достижна у реалним условима.
- Кориснику је важнија приближна процена него апсолутна тачност.

9.3. Избор и улога скалера (StandardScaler)

Пре уласка података у LSTM модел, вредности трошкова се нормализују помоћу StandardScaler-а. Овај поступак има за циљ да:

- Убрза процес учења,
- Смањи ризик од нестабилности током тренинга,
- Омогући моделу да ефикасније препозна временске обрасце.

Скалер се обучава над целокупним скупом дневних трошкова, а затим се користи и у фази предвиђања, како би скала података била конзистентна.

10. API функционалности

Бекенд део апликације реализован је као REST API сервис коришћењем FastAPI оквира. Основна улога API-ја је да омогући комуникацију између корисничког интерфејса и система за машинско учење, као и да обезбеди симулацију и предвиђање трошкова на основу историјских података.

API је дизајниран тако да подржава реалистичан начин рада, у коме систем свакодневно прави предвиђања, добија стварне податке и континуирано се прилагођава новим условима.

У наставку су описане кључне API руте које чине основу функционалности апликације.

Expense Predictor - Rolling Simulation

0.1.0 OAS 3.1

/openapi.json

default

POST	/simulateTomorrow	Simulatetomorrow
POST	/simulateTomorrowAuto	Simulatetomorrowauto
POST	/simulate	Simulate
GET	/simulate-json	Simulate Json
POST	/simulateTomorrowSingle	Simulatetomorrowsingle
POST	/simulateTomorrowAutoSingle	Simulatetomorrowautosingle
POST	/simulateSingle	Simulatesingle
GET	/simulate-json-single	Simulate Json Single

10.1. /simulate

Рута /simulate служи за покретање rolling walk-forward симулације над историјским подацима. Ова симулација имитира реалан сценарио у коме систем сваког дана предвиђа трошак за наредни дан, а затим упоређује ту прогнозу са стварним подацима из CSV фајла.

Основне карактеристике ове руте су:

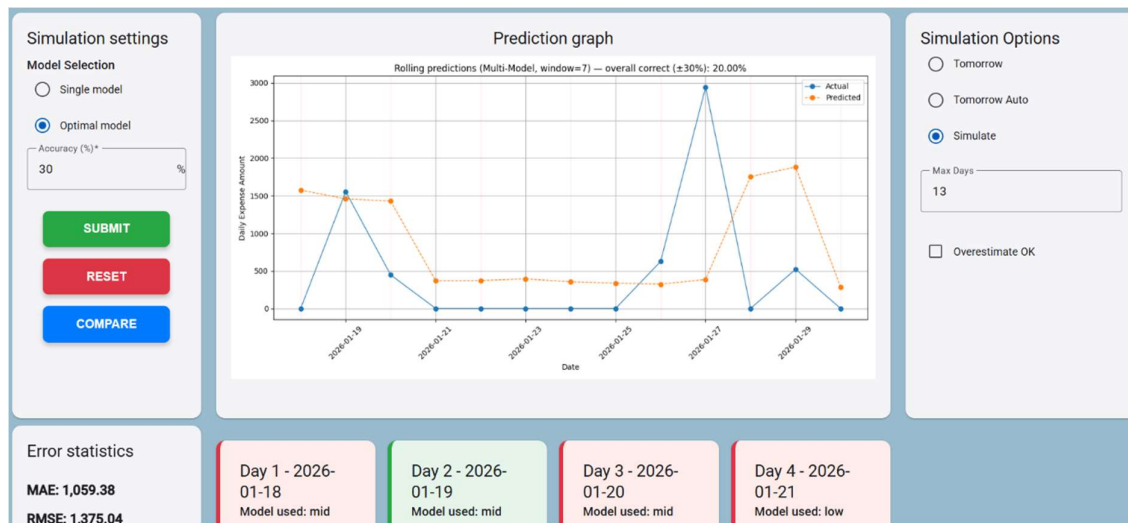
- Агрегација трансакција по дану (уколико постоји више трансакција у истом дану),
- Предвиђање укупног дневног трошка,
- Поређење предвиђене и стварне вредности,
- Континуирано „померање“ кроз време, дан по дан.

Резултат симулације укључује:

- Низ датума,
- Стварне дневне трошкове,

- Предвиђене дневне трошкове,
- Основне метрике тачности (нпр. MAE, RMSE).

Ова рута представља кључни механизам за тестирање понашања модела у условима који су веома блиски реалном коришћењу.



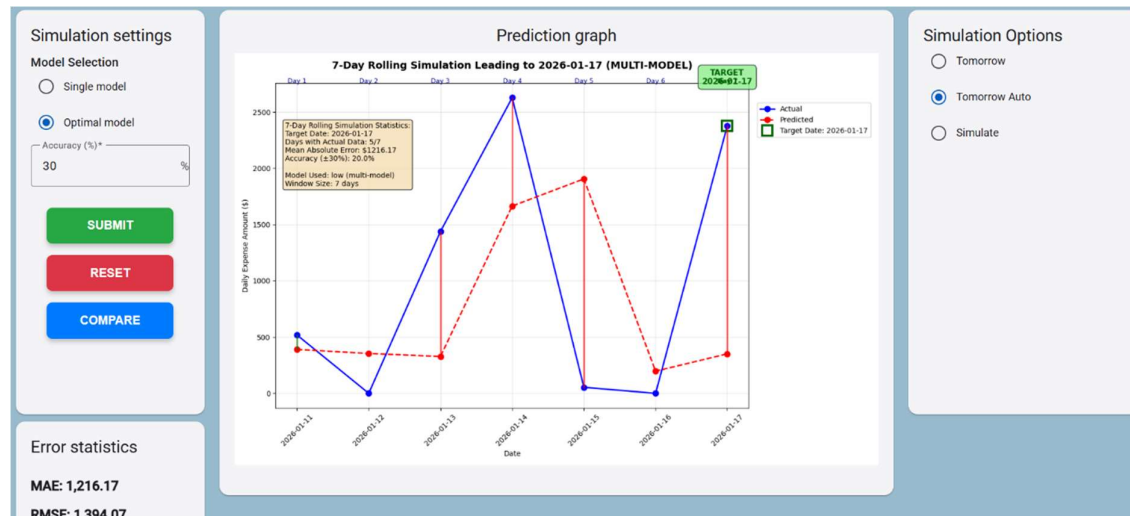
10.2. /simulateTomorrow

Рута /simulateTomorrow је намењена практичној, свакодневној употреби система. Њена улога је да на основу тренутног датума и доступних историјских података предвиди укупни трошак за наредни дан.

Кључне карактеристике ове руте су:

- Аутоматско преузимање тренутног датума са система,
- Коришћење последњих доступних дневних трошкова као улазног прозора,
- Избор одговарајућег LSTM модела (low, mid или high) у зависности од нивоа потрошње,
- Генерисање једне, јасне прогнозе за сутрашњи дан.

Ова функционалност представља директну примену целокупног система у реалном времену и најбоље илуструје његову практичну вредност за крајњег корисника.



11. Ограничења

Иако представљени систем показује да је могуће применити технике вештачке интелигенције за анализу и предвиђање личних финансијских трошкова, неопходно је јасно нагласити његова ограничења. Предвиђање финансијских расхода спада у домен сложених и високо варијабилних проблема, код којих апсолутна тачност није реално достижна.

Највећи изазов у оваквим системима лежи у разлици између статичких и динамичких трошкова. Статички трошкови, као што су кирија, рачуни за комуналне услуге, претплате и слични фиксни месечни издаци, имају релативно стабилан образац и понављају се у приближно истим износима. Овакви трошкови су погодни за предвиђање и модел их може научити са високом прецизношћу, нарочито ако постоји довољно историјских података.

Са друге стране, динамички трошкови представљају највећи извор неизвесности. У ову групу спадају спонтани трошкови као што су изласци, путовања, непланиране куповине, ванредни трошкови или промене животних навика. Овакви расходи не прате јасан временски образац и често су последица фактора који нису присутни у подацима (расположење, друштвени догађаји, економске околности и сл.). Због тога их није могуће поуздано предвидети само на основу историјских нумеричких вредности.

Додатно ограничење система представља чињеница да се модел ослања искључиво на историјске податке о износима трошкова, без додатног контекста као што су категорије, локације, тип дана (радни дан, викенд), празници или спољашњи економски фактори. Увођење оваквих додатних атрибута могло би потенцијално побољшати резултате, али би истовремено значајно повећало сложеност система.

На крају, важно је истаћи да резултати добијени овим системом треба да се посматрају као оријентациона помоћ, а не као апсолутна прогноза будућих трошкова. Сврха апликације није да са потпуном прецизношћу предвиди реалне издатке, већ да кориснику омогући бољи увид у сопствене финансијске навике и да послужи као подршка у планирању и оптимизацији месечног буџета.

12. Могућа унапређења

Иако тренутна верзија система успешно демонстрира примену вештачке интелигенције у анализи и предвиђању дневних трошкова, постоји више праваца у којима се решење може додатно унапредити и проширити. Ова унапређења би могла довести до прецизнијих предвиђања, боље прилагођености кориснику и шире практичне примене система.

Једно од најзначајнијих могућих унапређења јесте укључивање прихода као додатног улазног параметра. У тренутној имплементацији модел се ослања искључиво на историјске податке о трошковима, без увида у расположиви буџет корисника. Увођењем података о месечним или дневним приходима било би могуће реалније сагледати периоде у којима је повећана вероватноћа већих трошкова, као и идентификовати фазе у којима је потребна већа финансијска дисциплина. На тај начин би систем прешао са простог предвиђања износа на ширу анализу финансијске одрживости.

Додатно, значајно унапређење може се постићи увођењем временских и сезонских фактора. Трошкови корисника често зависе од годишњег доба, празника, почетка или краја године, као и од специфичних периода као што су летњи одмори или зимски празници. Увођењем информација о месецу, недељи у години, радним данима и викендима, као и празницима, модел би могао да научи дугорочне трендове и периодичне обрасце који нису очигледни из самих нумеричких података.

Још један потенцијални правац развоја јесте анализа трендова и навика корисника кроз време. Уместо да се систем ослања искључиво на краткорочни временски прозор (window size), могуће је комбиновати краткорочне и дугорочне трендове, чиме би се добио стабилнији и робуснији модел. Оваквим приступом систем би могао да препозна

постепене промене у понашању корисника, као што су промена животног стила или финансијских навика.

Са технолошке стране, постоји простор за унапређење коришћених модела и алата. У будућности је могуће експериментисати са напреднијим архитектурама као што су Bidirectional LSTM, GRU мреже или чак Transformer модели, који су се показали веома успешним у моделовању временских серија. Такође, могуће је увести аутоматску оптимизацију хиперпараметара, као и online learning приступ, где би се модел временом прилагођавао новим подацима без потребе за потпуним поновним тренирањем.

На крају, систем се може проширити и у смислу корисничког искуства, кроз напредније визуализације, персонализоване препоруке и упозорења, као и интеграцију са другим финансијским алатима. На тај начин би апликација прерасла из експерименталног система за предвиђање у свеобухватан алат за подршку у управљању личним финансијама.

13. Закључак

У оквиру овог пројекта реализован је систем за предвиђање дневних трошкова коришћењем техника вештачке интелигенције и машинског учења, са посебним фокусом на анализу временских серија. Циљ пројекта био је да се испита у којој мери је могуће предвидети будуће расходе на основу историјских података и на тај начин омогућити кориснику боље разумевање сопствених финансијских навика и потенцијалних будућих издатака.

Током развоја система уочено је да је проблем предвиђања трошкова по својој природи веома сложен. Иако одређени трошкови имају јасан образац и могу се сматрати релативно статичким (као што су рачуни, претплате и редовни месечни издаци), велики део расхода припада динамичкој категорији и директно зависи од спољних фактора, навика, тренутних потреба и непредвидивих околности. Управо због тога, предвиђање трошкова са потпуном тачношћу није могуће, што представља једно од кључних ограничења оваквих система.

Како би се побољшала робусност решења, архитектура система је еволуирала од иницијалног приступа са једним моделом ка сложенијем систему са више модела. Увођењем три LSTM модела, специјализована за различите опсеге трошкова, постигнута је боља адаптација на варијабилност података и стабилнија предвиђања. Додатни

експеримент са већим бројем модела (седам) показао је да повећање комплексности не мора нужно довести до бољих резултата, већ да може унети додатну нестабилност и компликовати одржавање система. Овај закључак наглашава значај баланса између сложености модела и практичне применљивости.

Посебан допринос пројекта представља примена rolling walk-forward симулације, која омогућава да се систем понаша као у реалном времену – сваки дан се врши ново предвиђање, након чега се у модел уводе стварни подаци. Оваквим приступом омогућена је реалнија евалуација перформанси модела и боље разумевање његовог понашања у практичним условима.

Иако систем има јасна ограничења у погледу апсолутне тачности, резултати показују да он може пружити вредне индикативне информације о будућим трошковима, нарочито у смислу трендова и очекиваних опсега издатака. Управо у томе лежи његов највећи потенцијал – не као алат за прецизно финансијско планирање, већ као помоћно средство за боље разумевање и оптимизацију личних финансија.

Уз могућа будућа унапређења, као што су укључивање прихода, сезонских фактора и напреднијих модела, овај пројекат има значајан потенцијал за даљи развој и примену у реалним системима за финансијску аналитику и подршку одлучивању.

14. Литература

1. Goodfellow, I., Bengio, Y., Courville, A.

Deep Learning. MIT Press, Cambridge, 2016.

2. Bishop, C. M.

Pattern Recognition and Machine Learning. Springer, New York, 2006.

3. Hochreiter, S., Schmidhuber, J.

Long Short-Term Memory. Neural Computation, Vol. 9, No. 8, 1997.

4. Brownlee, J.

Deep Learning for Time Series Forecasting. Machine Learning Mastery, 2018.

5. Géron, A.

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

6. FastAPI Developers

FastAPI – Official Documentation

fastapi.tiangolo.com

7. McKinney, W.

Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 2010.

8. NumPy Developers

NumPy Documentation

9. Hunter, J. D.

Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 2007.

10. Python Software Foundation

Python Documentation

docs.python.org/3

11. Hyndman, R. J., Athanasopoulos, G.

Forecasting: Principles and Practice. OTexts, 2018.

12. Scikit-learn Developers

Scikit-learn User Guide

scikit-learn.org/stable

13. Wikipedia

Artificial Intelligence

en.wikipedia.org/wiki/Artificial_intelligence

14. Wikipedia

Machine Learning

en.wikipedia.org/wiki/Machine_learning

15. Wikipedia

Artificial Neural Network

en.wikipedia.org/wiki/Artificial_neural_network

16. Wikipedia

Recurrent Neural Network

en.wikipedia.org/wiki/Recurrent_neural_network

17. Wikipedia

Long Short-Term Memory

en.wikipedia.org/wiki/Long_short-term_memory

18. Towards Data Science

Time Series Forecasting – Articles and Tutorials

towardsdatascience.com

19. TensorFlow Developers

TensorFlow – Official Documentation

20. Keras Developers

Keras – Official Documentation